



(19) BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENT- UND  
MARKENAMT

(11) Übersetzung der  
europäischen Patentschrift

(97) EP 0 573 503 B.1

(10) DE 692 32 869 T 2

(51) Int. Cl. 7  
G 06 F 9/44

G 06 F 17/40  
G 01 R 13/04  
G 01 R 29/00  
G 01 R 13/34

(21) Deutsches Aktenzeichen: 692 32 869.6  
(86) PCT-Aktenzeichen: PCT/AU92/00076  
(88) Europäisches Aktenzeichen: 92 905 546.5  
(97) PCT-Veröffentlichungs-Nr.: WO 92/015959  
(25) PCT-Anmeldetag: 25. 2. 1992  
(97) Veröffentlichungstag der PCT-Anmeldung: 17. 9. 1992  
(97) Erstveröffentlichung durch das EPA: 15. 12. 1993  
(97) Veröffentlichungstag der Patenterteilung beim EPA: 11. 12. 2002  
(41) Veröffentlichungstag im Patentblatt: 4. 9. 2003

DE 692 32 869 T 2

(30) Unionspriorität:  
486991 28. 02. 1991 AU

(71) Patentinhaber:  
Associative Measurement Pty. Ltd., North Ryde,  
Neusüdwaies, AU

(72) Vertreter:  
Patent- und Rechtsanwälte Bardehle, Pagenberg,  
Post, Altenburg, Geissler, 81679 München

(84) Benannte Vertragsstaaten:  
AT, BE, CH, DE, ES, FR, GB, IT, LI, LU, NL

(73) Erfinder:  
WILLIAMS, Victor, Donald, Balmain, AU; KEEBLE,  
Brian, John, Neutral Bay, AU; OATES, David, John,  
Quakers Hill, AU; CAMPOS, Guillermo, Alejandro,  
North Rocks, AU

(54) WISSENSCHAFTLICHES EMULATORGERÄT

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

DE 692 32 869 T 2

05.03.02

=12

692 32 869 6 08

Associative Measurement PTY LTD

6. März 2003

S.17755 EP/DE AI/PS/bb

Die vorliegende Erfindung bezieht sich auf wissenschaftliche Instrumente und insbesondere auf einen Emulator, welcher die Funktionen verschiedener wissenschaftlicher Instrumente erlaubt, wenn sie verbunden sind, emuliert oder reproduziert zu werden, ohne eines Kaufes, einer Installation, einer Verbindung und eines Tests der verschiedenen wissenschaftlichen Instrumente zu bedürfen. In ihrer bevorzugten Form erlaubt die vorliegende Erfindung, dass Messungen gemacht werden und Messprozesse verwendet werden, welche nicht möglich waren, gemacht zu werden und verwendet zu werden mit den verbundenen Instrumenten.

#### HINTERGRUNDECHNIK

In vielen Handlungsfeldern, die eine wissenschaftliche Grundlage haben, werden wissenschaftliche Instrumente verwendet, um die Ausgaben verschiedener Vorrichtungen zu analysieren, aufzuzeichnen und zu überwachen. Solche Vorrichtungen schließen Spannungsmesser, elektrokardiographische (ECG) Vorrichtungen, Mikrofone und Druck, Temperatur, Durchflussraten und ähnliche Messwandler ein. Entsprechend werden solche wissenschaftlichen Instrumente im Bauingenieurwesen, in der Elektrotechnik, der Akustik, dem hydraulischen Ingenieurwesen, in chemischen Prozessen, im biomedizinischen Ingenieurwesen, usw. verwendet.

Eine große Zahl solcher wissenschaftlicher Instrumente ist im Allgemeinen erforderlich, um (die) gewünschten Messungen durchzuführen. Solche Instrumente schließen Generatoren ein für verschiedene Wellenformen (wie z.B. Sinus, Quadrat, Rampe und Dreieck), Signalverarbeitungsvorrichtungen, wie z.B. Differentiatoren, Integratoren, Filter, Multiplizierer, usw., Analysierer, wie z.B. solche, die

erforderlich sind, um die schnelle Fourier-Transformation durchzuführen, und verschiedene Aufzeichnungsvorrichtungen, wie z.B. ein Diagramm-Aufzeichner, ein Datenschreiber, ein Kathodenstrahl-Oszilloskop oder ein Übergangsaufzeichner.

Solche Instrumente oder Vorrichtungen sind verhältnismäßig teuer und so besitzt jede Forschungseinrichtung oder ähnliche Organisation nur eine beschränkte Anzahl solcher Vorrichtungen. Dementsprechend gibt es einen erheblichen Wettbewerb unter Personen oder Gruppen innerhalb solcher Organisationen, welche die Vorrichtungen zu benutzen wünschen. Wenn man die Vorrichtungen erhalten hat, um den vorgesehenen Betrieb durchzuführen, ist es notwendig, dass diese Vorrichtungen gemeinsam angeordnet, verbunden und getestet werden, um sicher zu stellen, dass die Verbindungen richtig sind. Nur wenn diese Prozedur durchgeführt worden ist, ist es dann möglich, den vorgesehenen Betrieb aufzunehmen.

Nach dem Stand der Technik ist bekannt, Instrumente bereit zu stellen, die im Wesentlichen weitfortgeschrittene Kathodenstrahl-Oszillographen sind. Ein solches Instrument wird unter dem Namen SUPERSCOPE von G.W. Instruments aus Summerville Ma. USA 02143 verkauft und erlaubt, Wellenformen durch eine im Wesentlichen einen Speicheroszillographen darstellende Vorrichtung aufzunehmen und ebenso auf einem APPLE (registriertes Warenzeichen) Macintosh (registriertes Warenzeichen) Computer anzuzeigen. Diese Vorrichtung scheint jedoch keine Instrumente zu emulieren, sondern sie nimmt lediglich die Ausgabe solcher Instrumente auf und speichert sie zur darauf folgenden Anzeige.

Es ist auch bekannt, Software zur Datenerfassung zu verwenden. Ein solches Programm, das von LABTECH aus Wilmington, Ma und San Francisco, Kalifornien, USA, angeboten wird, ist eine graphische Schnittstelle, die wiederum Signale annimmt von verschiedenen Hardware-Elementen, die sich außerhalb des Computers befinden. Die Software sammelt Daten von mehrfachen Kanälen, führt eine Analyse durch und, falls nötig, eine Reduktion der Daten und erzeugt Anzeigen.

Diese Aktivität ist in Echtzeit verfügbar. Es wird wiederum kein Versuch gemacht, die Funktion wissenschaftlicher Instrumente zu emulieren. Ein weiterer Unterschied besteht darin, dass die graphische Schnittstelle eine Anzahl verschiedener Maschinen miteinander verbindet, anstatt im Wesentlichen in eine Einheit integriert zu sein.

Ein Datenakquisitions- und Analysesystem, das unter dem Handelsnamen LABVIEW2 von National Instruments aus Austin, Texas, USA, gehandelt wird, ist ebenso bekannt. In diesem System werden Icons verwendet, um sowohl Datenerfassungsfunktionen als auch Datenanalysefunktionen darzustellen. Diese Icons werden verwendet, um ausführbaren Code zu erzeugen, der lediglich von einem Personalcomputer ausgeführt wird, in welchem die Software des Systems geladen ist. Dieses System verwendet keine Hardware, sondern lediglich Software. Die Datenerfassungs- und Analysefunktionen können sequentiell verbunden werden, so dass eine Dateneingabe in den Computer zuerst erfasst und dann analysiert wird. Schließlich werden die analysierten Daten dargestellt. Es sei bemerkt, dass der Computer eingesetzt wird, um die gewünschten Verbindungen zwischen externen Instrumenten herzustellen, aber dass er keine Instrumente emuliert oder ein Programm kompiliert, um die Signalverarbeitungsfunktionen einer Gruppe von miteinander verbundenen Instrumenten zu reproduzieren. Zum Beispiel wird keine Vorkehrung getroffen zur Rückkopplung von einem Icon zurück zu einem anderen im Sinne einer Steuerung. Die niedrigen Verarbeitungsgeschwindigkeiten und die unbestimmte Natur der Multitasking-Umgebung des APPLE (eingetragenes Warenzeichen) Personalcomputers erfordern die Bereitstellung einer Zeit-"Koordinate" zur Begleitung der erfassten Daten.

Dieses System ist ein virtuelles Instrument, d.h. es ist lediglich durch eine Software-Simulation gekennzeichnet und muss im Gegensatz gesehen werden mit dem neuen Konzept einer Emulation. Emulation umfasst flexible, mehrfach verwendbare, wiederprogrammierbare Hardware, die Code ausführt, der von einem

09.03.02

-4-

graphischen Compiler bei hoher Verarbeitungsgeschwindigkeit abgeleitet ist, der Echtzeit-Instrumentenemulation erlaubt.

US 4.315.315 bezieht sich auf einen Prozess zur automatischen Erzeugung eines  
5 Computerprogramms in einer Maschinen-Assembler-Sprache unmittelbar aus  
einem zweidimensionalen Netzwerk, welches den Datenfluss und die Steuerlogik  
darstellt, die verlangt wird, um auf einem angegebenen, allgemein verwendbaren  
Digitalrechner ausgeführt zu werden. Eine Emulation wird jedoch nicht durchge-  
führt.

10

Der Artikel "Using Mathematica in Support of LabView: Power in the Laboratory",  
NORTHCON Conf. Rec. Seattle, WA, USA, 9-11. Okt. 1990, Seiten 353-  
358, bezieht sich auf eine Graphik-orientierte Computersprache zum Erzeugen  
von Programmen, welche auf einem Personalcomputer ausgeführt werden. Es  
15 wird keine Emulation durchgeführt.

Deshalb ist die Aufgabe der vorliegenden Erfindung, einen wissenschaftlichen  
Instrumenten-Emulator bereit zu stellen, und ein entsprechendes Verfahren, wel-  
che die Funktion von wissenschaftlichen Instrumenten reproduzieren. Diese Auf-  
20 gabe wird gelöst durch den Emulator gemäß Anspruch 1 bzw. durch das Verfah-  
ren gemäß Anspruch 8. Weitere Ausführungsformen sind in den abhängigen An-  
sprüchen bestimmt.

Die Aufgabe des Emulators ist es, mit Hilfe einer Kombination von sowohl  
25 Hardware als auch Software die Funktionen verschiedener wissenschaftlicher In-  
strumente zu reproduzieren, um damit nicht nur individuelle Funktionen von In-  
strumenten zu erlauben, die reproduziert werden sollen, sondern erlauben auch,  
dass solche reproduzierten Funktionen miteinander verbunden werden können, so  
dass der Emulator als Ganzes die gesamte Funktion eines Feldes, das aus mitein-  
30 ander verbundenen wissenschaftlichen Instrumenten gebildet wird, reproduzieren  
kann.

Eine Anzahl von sehr wesentlichen Vorteilen wird auf diese Weise erreicht. Zuerst werden die Kosten der Vorrichtungen wesentlich reduziert, da die einzelnen wissenschaftlichen Instrumente, deren Funktionen reproduziert werden sollen, nicht angeschafft werden müssen. Zweitens wird auch eine wesentliche Ersparnis in der Einrichtungszeit erreicht, da es nicht notwendig ist, dass solche Instrumente von verschiedenen Orten eingekauft und an einem Punkt zusammengebaut werden müssen und physikalisch verbunden werden müssen mit Kabeln, Steckern, Adaptern, usw. Drittens werden die erzielten Ergebnisse dadurch sehr verbessert, dass die Ausgabe der Gruppe von miteinander verbundenen wissenschaftlichen Instrumenten nicht nur in Echtzeit angezeigt werden kann, sondern sie zusätzlich in Echtzeit verarbeitet wird und sie kann ebenso gespeichert werden zur nachfolgenden Manipulation, Analyse und Bewertung. Schließlich können die Ergebnisse von gleichzeitigen oder parallelen Messungen leichter aufeinander bezogen werden, so dass die Zwischenbeziehungen zwischen Messungen gesichtet werden können, insbesondere in komplexen Systemen.

Die Erfindung ist dargelegt im angefügten Vorrichtungsanspruch 1 und im Verfahrensanspruch 8.

Vorzugsweise sind die Instrumenteneinheiten jeweils dargestellt durch einen entsprechenden Icon, der angezeigt werden kann auf dem Videoanzeigebildschirm während des Betriebs des Installationsprogramms, um es dem Operator zu erlauben, die gewünschte Instrumentationseinheit auszuwählen und in dem Feld zu platzieren.

Der Compiler erzeugt im Speicher eine Anzahl von verschiedenen Programmen, welche das Feld von Instrumentationseinheiten darstellen. Diese Programme sind auf dem Prozessor im Computersystem verteilt, um das gewünschte Feld von Instrumentationseinheiten zu emulieren. Vorzugsweise sind die vom Compiler erzeugten Programme im Speicher gespeichert, um vorkonfigurierte Instrumentati-

onsfelder zu erzeugen, welche sofort in der Lage sind, gewünschte Verarbeitungsfunktionen auszuführen. Diese Programme speichern das akkumulierte intellektuelle Werk des Benutzers.

5 Vorzugweise ist die Anzahl von Prozessoren, Videogeneratoren und Analogsignalmodulen erweiterbar, um den gewünschten Grad von Komplexität und/oder Verwendbarkeit des Feldes von Instrumentationseinheiten zu erreichen.

10 Dem Analogsignalmodul steht ein elektrisches Echtzeitsignal zur Verfügung, welches verwendet werden kann, um andere Hardwareelemente zu betreiben oder anzusteuern. Vorzugweise stellt es sowohl einen analogen als auch digitalen Ausgang bereit.

15 Die Ein-/Ausgabe des wissenschaftlichen Instrumentenemulators ist vorzugsweise auch zugänglich von anderen Ressourcen, welche innerhalb des Computers verfügbar sind, einschließlich Netzwerk-Kommunikationsschnittstellen (RS232, ETHERNET, etc.) und Busschnittstellen, wie z.B. IEEE-488-GPIB, ISA und EISA. Diese Ressourcen können unabhängig bereit gestellt werden durch den Käufer oder OEM's (original equipment manufacturers).

20 Die Daten, welche in dem Speichermittel gespeichert sind, stehen vorzugsweise auch für den Daten-Export zur Verfügung zu verschiedenen Standard-Computerpaketen, wie denjenigen, die unter den Handelsmarken EXCEL, LOTUS und AXUM verkauft werden, wobei die angesammelten Daten manipuliert werden können für eine nachfolgende graphische Darstellung und zur Tabellierung, um die Berichterzeugung zu vereinfachen.

30 Gemäß eines zweiten Gesichtspunkts der vorliegenden Erfindung wird ein wissenschaftliches Instrument offenbart zum Messen und Aufzeichnen elektrischer Wellenformen, wobei das Instrument einen Computer umfasst, der eine Zentraleinheit und elektronische Speichermittel aufweist, welche in einem Gehäuse

untergebracht sind, wobei das Gehäuse die Bereitstellung von mindestens einem Floppy-Disk-Schacht umfasst und eine Vielzahl elektrischer Verbinder, die untergebracht sind in dem Raum, der vorgesehen ist für einen Floppy-Disk-Schacht, wobei die elektrischen Verbinder verbunden sind mit dem Speichermittel.

5

Gemäß eines dritten Gesichtspunktes der vorliegenden Erfindung wird ein Kompilierungsverfahren offenbart zum Erzeugen von Objektcode, um die mathematische/Signalverarbeitungs-Prozedur eines elektrischen Schaltkreisfunktionsblocks zu implementieren, der mindestens eine Eingabe aufweist, um daraus eine Ausgabe zu bilden, wobei das Verfahren die Schritte umfasst der Darstellung der Funktion als eine Folge elementarer mathematischer Schritte, von denen jeder selbst unmittelbar darstellbar ist in dem Objektcode, und Anordnen der Objektschritte in eine Folge zur sequentiellen Ausführung, beginnend mit der Eingabe/den Eingaben. Vorzugsweise kann ein Parameter des Funktionsblocks spezifiziert werden. Ebenso offenbart ist ein Kompilierungsverfahren zum Erzeugen von Objektcode, um die mathematischen/Signalverarbeitungsfunktion eines elektrischen Schaltkreises zu implementieren, welche mindestens eine Eingabe und mindestens eine Ausgabe aufweist und gebildet wird durch Verbindung einer Vielzahl von funktionalen Blöcken, von denen jeder eine mathematische/Signalverarbeitungsfunktion aufweist, für welche ein Objektcode kompiliert wurde in Übereinstimmung mit dem Obigen, wobei das Verfahren die Schritte aufweist der Darstellung der elektrischen Schaltkreis-mathematischen/Signalverarbeitungsfunktion als Folge von Ereignissen, von denen jedes darstellbar ist in dem Objektcode, und Anordnen der Objektschritte in einer Folge zur sequentiellen Ausführung, beginnend mit der Eingabe/den Eingaben.

#### BESCHREIBUNG DER ZEICHNUNGEN

30 Eine Ausführungsform der vorliegenden Erfindung wird nun beschrieben werden, unter Bezugnahme auf die Zeichnungen, in welchen:



05.03.02

-8-

Fig. 1 ein schematisches Blockdiagramm der Hardwaremodifikationen ist, die an einem IBM-PC erforderlich sind,

Fig. 2 ein Blockdiagramm des Analogmoduls von Fig. 1 ist,

Fig. 3 ein Schaltkreisdigramm von einem von identischen Relais von Fig. 2 ist,

Fig. 4 eine schematische Karte der Speicheranordnung innerhalb des Computers ist,

Fig. 5 ein Blockdiagramm der Echtzeit-Video-gedruckten Schaltung von Fig. 4 ist,

Fig. 6 eine Bildschirmanzeigenauflistung der Icons ist, die verschiedene Instrumentationseinheiten darstellen, die innerhalb des Bibliotheksprogramms gespeichert sind,

Fig. 7 ein typisches Feld ist, welches durch Verbindung der verschiedenen Instrumentationseinheiten gebildet wird,

Fig. 8 ein Blockdiagramm eines vergleichsweise einfachen Feldes ist, das nützlich ist in biomedizinischen Anwendungen,

Fig. 9 eine Wiedergabe der Bildschirmanzeigefenster ist, die dem Feld von Fig. 8 entsprechen,

Fig. 10 ein Feld ist, das eine lineare Interpolation bildet,

Fig. 11 die Ausgabe der drei Bildschirmanzeigen zeigt, die in Fig. 10 veranschaulicht sind,

Fig. 12 ein Feld ist, welches eine phasenverriegelte Schleife bildet,

Fig. 13 die Ausgabe von den drei Bildschirmanzeigen, veranschaulicht in Fig. 12, zeigt.

Fig. 14 ein Feld ist, welches eine analoge Lösung bereit stellt für eine Differenzialgleichung zweiter Ordnung.

Fig. 15 die Ausgabe der beiden Bildschirmanzeigen, veranschaulicht in Fig. 14, zeigt, und.

Fig. 16 ein komplexeres Feld ist.

Die Anhänge I-IV listen verschiedenen Programmfragmente auf, die nachfolgend beschrieben werden.

#### BESCHREIBUNG DER BEVORZUGTEN AUSEFÜHRUNGSFORM

Es sei jetzt auf Fig. 1 Bezug genommen, worin die Vorrichtung der bevorzugten Ausführungsform ganz enthalten sein kann innerhalb des Gehäuses 1 eines herkömmlichen IBM (eingetragenes Warenzeichen) Personal Computers, der einen ISA oder EISA Bus aufweist auf der Basis des ursprünglichen IBM-AT. Innerhalb des Gehäuses 1 sind die üblichen Komponenten untergebracht von einer Zentraleinheit (CPU) 2, einem Speicher 3 und einem 8MHz-Bus 4.

Innerhalb eines Reserve 5 1/4-Zoll-Floppy-Disk-Einschubs ist ein Analogmodul 6 untergebracht, auf welchem bestimmte Eingabestecker 7, bestimmte Ausgabestecker 8 und allgemeine Ein-/Ausgabeleitungen 5 für Verstärker, Frequenzzähler, Abtastaktsynchronisierung, digitale Eingaben und Ähnliches montiert sind.

09.03.02

-10-

Auf dem Bus 4 befinden sich 4 Schlitze für gedruckte Schaltkreiskarten 9. Die 4 PC-Karten 9 sind bezeichnet mit A, B, C bzw. D. Die drei benachbarten PCBs A bis C einschließlich stellen jeweils einen Master-PCB, einen Slave-PCB und einen Video-PCB dar. Der Video-PCB treibt seinerseits eine bekannte VGA-gedruckte Schaltkreiskarte D, die von Tseng Labs verkauft wird, welche eine 800 x 600 Auflösung in 256 Farben bereit stellen kann. Diese PC-Karte D wird direkt verbunden mit dem Videoanzeigeschirm 10.

Innerhalb des Analogmoduls 6 befinden sich folgende Systemressourcen:

1 VIER isolierte (optionale) analoge i/p-Kanäle. Jeder Kanal weist eine programmierbare 9-120dB Verstärkungsleistung auf (3 Mikrovolt Auflösung, 0 Signal-zu-Rauschverhältnis von eins), programmierbare Anti-Alias-Filtrierung und eine ADC-Umwandlung mit 12-Bit-Auflösung. Jeder Kanal kann AC oder DC gekoppelt sein mit langen AC-Kopplungszeitkonstanten (2 Minuten) und weist unabhängige Steuerungen für AC oder DC-Offsets auf, die gesteuert werden können von den Laufzeitbildschirmen. Die Abtastrate kann 15 KHz pro Kanal (abhängig von der Projektverarbeitungslast) betragen und die Zahl der Analogmodule, die mit der gleichen Slave-Prozessorkarte verbunden sind. Die Eingaben sind auf 3,5 KV Gleichstrom isoliert.

2 ZWEI Analogausgaben mit einem Spannungsbereich von +/- 10 V und einer Stromstärke von +/- 100 mA. Diese können verwendet werden zur Vorspannung von Spannungsmessern (AC oder DC-getrieben), Steuerausgaben, etc.

3 VIER auswählbare analoge Ausgaben von hohem Niveau, eine von jedem der obigen Verstärker. Diese treiben digitale FM-Band-Recorder an, um selten auftretende Ereignisse zu speichern zur Wiedergabe in dem Prozessor (2).

ODER

09-03-02

-11-

VIER auswählbare Eingaben von hohem Niveau zu jedem der obigen Verstärkerkanäle. Das System wird in diesem Modus geschaltet zum Wiedergeben von Ereignissen, welche im Ausgabemodus auf Band genommen wurden.

4 ACHT Bits einer Masse-referenzierten digitalen Eingabe.

5 ACHT Bits einer digitalen Ausgabe, die verwendet werden kann für Relais-Treiber oder Ereignisindikatoren.

6 EINE Frequenzgeneratorausgabe (Faktgenerator) 0-2 MHz 0,1% Genauigkeit.

7 EIN Ereigniszähler/Frequenzzähler. Eingabe 0,1 Hz - 8MHz.

8 EINE 5-Volt Referenz 100 mA  $\pm$  5% (Für Spannungsmesser, etc.).

9 EINE Abtasttaktausgabe Referenzleitung zum Synchronisieren einer Abtastung zwischen mehreren wissenschaftlichen Instrumentenemulatoren der bevorzugten Ausführungsform.

10 EINE Abtasttakteingabe Referenzleitung zum Synchronisieren einer Abtastung von einem "Master"-wissenschaftlichen Instrumentemulator (zur Verwendung mit "Slave"-Emulatoren), und.

11 FÜNF Massekabel.

Das Analogmodul 6 und die PC-Karten 9 sind alle durch verschiedene Hilfsbusse 11, 12, 13 bzw. 14 verbunden.

Wie in Fig. 2 zu sehen, wird das Analogmodul 6 von Fig. 1 bereit gestellt mit vier analogen Ein-/Ausgabeverbindern 20, vier analogen Eingaben 21, zwei analogen

Ausgaben 22, einem Frequenzausgabezehlereingang 23, einem Taktoutput 24, einem 8-Bit-digitalen Eingang 25, einem 8-Bit-digitalen Output 26, einer 5 Volt-Referenzspannung 27 und einem Slave-Synchronisieroutput 28.

5 Jeder der analogen Eingänge 21 ist verbunden über einen Frontendverstärker 31 mit einem Isolator 32, dessen Output verbunden ist mit einem Relais 33. Das Relais 33 ist auch mit den analogen Ein-/Outputverbindern 20 und einem Verstärker 34 verbunden, welcher eine programmierbare Verstärkungsleistung, eine AC/DC-Kopplung aufweist und einen AC-corner und einen DC-Offset aufweist.  
10 Die Output des Verstärkers 34 wird ihrerseits zu einem Abtast- und Halteschaltkreis 35 geleitet, dessen Output empfangen wird von einem analogen Multiplexer 36. Die Output des Multiplexers 36 wird über einen A/D-Wandler 37 zu dem Hilfsbus 11 geleitet, der die analogen Module 6 und die PC-Karte 9B verbindet.

15 Der Betrieb der Verstärker 34 und der Abtast- und Halteschaltkreise 35 wird gesteuert von einer digitalen Steuereinheit, einem Adress-Dekodierer und einem A/D-D/A-Sequenzierer 39, welcher sowohl Daten vom Bus 11 als auch Abtasttakt- und Sequenziertaktsignale empfängt. Die Steuereinheit/Dekodierer/Sequenzierer 39 gibt auch über D/A-Wandler 40 zu den analogen  
20 Ausgängen 22 über einen Outputverstärker 41 aus.

Der Frequenzzehlereingang 23 bzw. Taktoutput 24 kommunizieren direkt mit einem Zähler 42, der seinerseits direkt mit dem Hilfsbus 11 kommuniziert.

25 Jeder der digitalen Eingänge 25, der digitalen Ausgänge 26, Referenzspannung 27 und des Slave-Synchronisieroutputs 28 ist verbunden mit einem digitalen Ein-/Outputschaltkreis 43, der seinerseits direkt mit dem Hilfsbus 11 verbunden ist.

30 Die Taktanordnungen des Schaltkreises, der in Fig. 2 veranschaulicht ist, sind in zwei Sequenzen geteilt. Die erste Sequenz betrifft die digitale Eingabe und Aus-

gabe. Wenn es vom Programm verlangt wird, wird diese digitale Eingabe und Ausgabe in Gang gesetzt durch individuelle Befehle von einer im Wesentlichen herkömmlichen Datenerfassungssteuereinheit, die einen Teil des Slave-Prozessors auf PCB 9B (Fig. 1 und Fig. 4) bildet.

Die zweite Sequenz ist der Fluss digitaler Daten, die umgewandelt wurden von analogen Eingaben oder die umgewandelt werden sollen, um analoge Ausgaben bereit zu stellen. Diese digitalen Daten werden empfangen und abgeschickt unter der Steuerung der Steuereinheit/Dekodierer/Sequenzierer 39, die voreingestellt werden kann, um die erforderliche Zahl ankommender oder abgehender analoger Kanäle zu betreiben. Der Controller/Dekodierer/Sequenzierer 39 führt einen vollständigen Zyklus des Eingebens und Ausgebens oder einer Sequenz in jeder Abtastperiode aus und er führt dies mit minimalem Prozessoraufwand durch und erhöht auf diese Weise die Betriebsgeschwindigkeit des Datenerfassungs-Controllers, auf den oben Bezug genommen wurde auf dem Slave-Prozessor von PCB 9B.

Andere Funktionen des Schaltkreises von Fig. 2, wie z.B. die Frequenz, die ausgegeben werden soll als die Taktausgabe 24, der Bereich der Frequenz, der gezählt werden soll von Frequenzzählereingang 23, und irgendein Synchronisationssignal, welches erforderlich ist für das Slave-Synchronisationssignal 28, werden eingestellt zu Beginn der Ausführung des graphischen Compiler-Programms durch geeignetes Angeben des entsprechenden Icons.

Fig. 3 veranschaulicht im Detail die Natur des Relais 33, das zwischen zwei Positionen geschaltet werden kann. In der gezeigten Position wird die Eingabe vom analogen Eingang/Ausgang 20 zum Verstärker 34 geführt, womit die analogen Eingänge 21 isoliert sind. In der alternativen Stellung wird der Eingang von den analogen Eingängen 21 zu dem Verstärker 34 geführt, aber er wird auch als Ausgang verfügbar gemacht am analogen Eingang/Ausgang 20.

Fig. 4 ist eine bildhafte Darstellung der Prozessorarchitektur, die eine Speicherkarte beinhaltet, der Host-Computer 22 mit seinem zugeordneten Speicher 3 ist verbunden mit dem Video-PCB 9C, welches seinerseits mit dem Slave-PCB 9B verbunden ist.

Wie angegeben in Fig. 4, können bis zu 8 analoge Module 6 verbunden werden zu jeder PCB 9B und bis zu 4 Slave-PCBs 9B können hinzugefügt werden.

Die Speicherkarte ist im Wesentlichen dreidimensional, wobei der Speicher 3 des Host-Computers 2 überlappt und damit zugänglich ist mit den anderen Abschnitten des Schaltkreises, die den Speicher verwenden.

Ein Blockdiagramm des Video-PCB 9C ist veranschaulicht in Fig. 5. In diesem Diagramm sind die Verbindungen zwischen dem herkömmlichen Videografikadapter (VGA) des Host-Computers hergestellt über den herkömmlichen VGA-Verbinden 50. Die Verbindungen zwischen dem Host-Computer 2 und dem Video-PCB 9C werden über Bus 4 geführt. Ebenso werden die Verbindungen zwischen dem Video PCB 9C und jedem der Slave-PCBs 9B über Bus 13 geführt wie oben angegeben in Fig. 1.

Daten, die entweder vom Bus 4 und/oder 13 empfangen werden, werden über einen logischen Dual-Port-Schaltkreis 51 zu einem Zuerst-ein-zuerst-aus (FIFO)-Puffer 52 geführt. Der FIFO 52 gibt an einen logischen Feldschaltkreis 53 aus, der drei getrennte wahlfreie Speicher aufweist, einen horizontalen RAM 54, einen vertikalen RAM 55 und einen statischen RAM 56.

Der Ausgang des logischen Felds 53 wird über den Vergleich 57 zu einem Videogenerator 58 geführt und dann zu dem VGA-Verbinden 50. Zusätzlich empfängt das logische Feld 53 auch drei Signale von dem VGA-Verbinden in Form von horizontalen Synchronisationspulsen, vertikalen Synchronisationspulsen und einem Punkt Takt.

Im Wesentlichen nimmt das logische Feld 53 die Daten, die über Busse 4 und/oder 13 geliefert werden, auf und berechnet Pixel, um die einzelnen Pixel zu ersetzen, die von dem Videografikadapter (VGA) 12 erzeugt werden, und es stimmt mit Multimedia-Standards überein zur Kommunikation des Videobildes auf Bus 14.

Der Speicher 3 (Fig. 1 und 4) weist eine Bibliothek von Instrumentationseinheiten auf. Davon wird jede dargestellt durch einen Icon, und Fig. 6 liefert einen Eindruck des Bereichs von Instrumentationseinheiten, die ausgewählt werden können von einer gegebenen Bibliothek. Durch Verwendung einer Maus in bekannter Art kann der Operator ausgewählte Instrumentationseinheiten verbinden aus der Icon-Liste von Fig. 6, um ein Feld zu bilden von miteinander verbundenen Instrumentationseinheiten, veranschaulicht in Fig. 7. Während der Erzeugung des Felds von Fig. 7 überprüft die Maschine unter der Steuerung von Software, dass das Feld keine nicht verbundenen Eingaben, logisch nicht akzeptierbare Verbindungen und ähnliche Defekte aufweist. Jeder detektierte Effekt wird angezeigt.

Nachdem das Feld verbunden wurde zur Zufriedenheit sowohl des Operators als auch des Setup-Programms, das während dieser Phase verwendet wird, wird dann ein Compiler-Programm zum Ablauf gebracht, welches aus der graphischen Darstellung des Feldes ausführbaren Objektcode erzeugt, der die gesamte Signalverarbeitungsfunktion des gesamten Felds ausführt. Als Folge davon, wenn in Echtzeit das Eingangssignal angewandt wird auf das Feld, wird/werden das eingehende Signal/die eingehenden Signale manipuliert und der eine oder mehrere Ausgänge des Feldes werden angezeigt in Echtzeit auf den Videofenstern, die angezeigt werden können auf dem Schirm 10, gespeichert auf Platte, usw.

Wesentlich für das Verständnis der Art, in welcher das Compiler-Programm arbeitet, ist eine Würdigung, dass jeder Icon selbst ein Minifeld darstellt, das aufgebaut werden kann von sehr fundamentalen Schritten, die ihrerseits leicht von dem



Programm ausgeführt werden können. Wenn z.B. die Grundschrirte angenommen werden, Addition und Subtraktion zu sein, dann kann eine Modifikation betrachtet werden als wiederholte Addition und eine Division kann betrachtet werden als wiederholte Subtraktion. Mit diesem Hintergrund vor Augen kann es gewürdigt werden, dass ein Rampenspannungsgenerator erzeugt werden kann von einem Null-Anfangsniveau durch sukzessive Addition von sehr kleinen Inkrementen, bis ein vorbestimmtes Niveau erreicht wird. Dann wird dieses Niveau selbst subtrahiert, um den Null-Startpunkt wieder zu erzeugen. Dann werden die kleinen Additionen wiederum durchgeführt, usw.

Eine Berücksichtigung der verschiedenen Icons, die in Fig. 3 dargestellt sind, wird ebenso auf die Würdigung führen, dass der Icon selbst im Wesentlichen eine graphische Form jeder gewünschten Gestalt darstellt, die entworfen ist, um eine Darstellung einer bestimmten mathematischen oder Signalverarbeitungsfunktion zu überbringen, die sowohl vom Benutzer gesehen werden kann als auch identisch verstanden wird vom graphischen Compiler-Programm. Die Gestalt enthält mindestens einen Eingabeknoten und/oder mindestens einen Ausgabeknoten, die jeweils Eingangs- und Ausgangspunkte für Datenströme darstellen. Zum Beispiel wird ein Addierer, der Eingabeknoten A und B und einen Ausgabeknoten C aufweist, kompiliert, um das Programm  $C = A + B$  zu ergeben. Diese Icon-Elemente, welche lediglich einen oder mehrere Ausgangsknoten aufweisen, werden Quell-Icon-Elemente genannt (z.B. ein Spannungsgenerator), während diejenigen Icons, die lediglich einen oder mehrere Eingabeknoten aufweisen, ein Ziel-Icon genannt werden (z.B. ein Anzeigefenster).

Ein verbindender Pfad, der irgendeinen Ausgabeknoten verbindet und in einem Eingabeknoten endet, wird ein Datenstrom genannt. Der Datenstrom trägt den Datentyp, der mit dem Ausgabeknoten assoziiert wird, und als Folge muss der Eingabeknoten, in welchem er endet, vom gleichen Typ sein. Dies stellt eine zusätzliche Regel dar, welche dem Compiler-Programm eigen ist. Da die Verbindungen zwischen Knoten Datenströme repräsentieren anstelle einer physikali-

sehen Verdrahtung ist es zulässig, dass die Verbindungen zwischen Icons sich mit anderen Verbindungen oder selbst mit anderen Icons ohne eine nachteilige Wirkung kreuzen. Das führt daher, dass der Datenstrom einen Ausgabeknoten und einen Eingabeknoten aufweist und nicht von irgendeiner dazwischen liegenden Stelle beeinflusst wird. Die Koordinaten der Quellen- und Zielknoten auf einem beliebigen Koordinatensystem werden verwendet, um Software-"Zeiger" zu erzeugen zu Quell- und Zieldatenpuffern zur Ausführung der Signalverarbeitungsfunktionen auf den Daten, auf die "gezeigt" wird.

- 10 Ebenso beinhaltet von einem Icon wird eine Icon-Spezifikation, bei der der Icon selbst anzureichend ist, um die gesamte Funktion des Icons zu beschreiben. Zum Beispiel hat ein Verstärker eine Funktion,  $\text{Ausgang} = G \times \text{Eingang}$ , wobei  $G$  die Verstärkungsleistung des Verstärkers darstellt. Um jedoch die Verstärkungsleistung, die vom Benutzer angegeben ist, zu aktivieren, kann die Verstärkungsleistung 15 eingegeben werden als vorbestimmter Parameter mittels der Icon-Spezifikation.

- Es wird gesehen werden, dass unter Verwendung der obigen Vorgehensweise eine Bibliothek von vorbereiteten Icons vorbereitet werden kann, mit ihrem eigenen Programm, um die mathematische/signalverarbeitende Funktion des Icons auszuführen. Außerdem kann dieselbe Vorgehensweise wieder angewandt werden, wenn es 20 wünschenswert ist, ein Programm zu kombinieren, das mathematische/Signalverarbeitungsfunktionen ausführen soll auf einem Feld, das von miteinander verbundenen Icons gebildet wird. Um ein einfaches Beispiel von einem 25 Feld anzunehmen, sei ein Feld betrachtet, welches zwei Eingaben  $A$  und  $B$  und eine Ausgabe  $C$  aufweist, wobei das Feld aus einem Addierer besteht, der Eingänge  $A$  und  $B$  aufweist und der Ausgang des Addierers verbunden ist mit einem Verstärker, der eine Verstärkungsleistung  $G$  aufweist, wobei der Ausgang des Verstärkers den Ausgang des Feldes darstellt. Die mathematische/signalverarbeitende Funktion ist  $C = G \times (A + B)$ . Der graphische Compiler 30 der bevorzugten Ausführungsform erzeugt Maschinencodes auf dieselbe Weise

wie es ein FORTRAN-Compiler tun würde, wenn er im Wesentlichen dasselbe Statement erhält in Source-Code, geschrieben in der FORTRAN-Sprache.

Anhang I stellt ein Programmfragment dar von Code, der es dem Benutzer erlaubt zu entscheiden, was die Icon-Spezifikation sein wird für ein gegebenes Icon. Der „Pop-up“-Fensterabschnitt der Anzeige, in welchem der Benutzer den numerischen Wert/die numerischen Werte, die spezifiziert werden sollen, eingibt, wird ein „Blatt“ genannt.

10: Ebenso stellt das Programmfragment von Anhang II ein Beispiel dar von einem Programm, welches verwendet wird zum Kompilieren eines Icons. Der Code, der beim Kompilieren eines Feldes verwendet wird, ist ähnlich.

Anhang III ist ein Codefragment, welches ein Beispiel darstellt von einem digitalen Signalverarbeitungs-(DSP)-Implementationscode. Dieser Code erlaubt, dass digitale Signale, welche die Ausgabe von tatsächlichen elektronischen Vorrichtungen replizieren (wie dargestellt durch ein Icon oder ein Feld), erzeugt werden.

Schließlich stellt Anhang IV ein Codefragment dar von der Zeitgabe oder der Sequenz, die in der digitalen Signalverarbeitung verwendet werden. Dies stellt die zeitgerechte Vollendung der Berechnungen sicher in der Zeit zwischen aufeinanderfolgenden Abtastungen.

25: Der Betrieb in „Echtzeit“ wird unterstützt durch die Art, in welcher analoge Eingangssignale erfasst werden. Der zulässige  $\pm 10$  V Eingabebereich wird dargestellt durch eine 12-Bit-Zahl und das analoge Eingangssignal wird abgetastet mit einer Abtastfrequenz von 1 bis 60.000 Punkten oder Abtastungen pro Sekunde. Die genaue Abtastfrequenz wird angegeben oder ist auswählbar vom Benutzer.

Als Folge dieses Abtastschemas müssen alle Programmierschritte, die erforderlich sind, um eine bestimmte Funktion eines Icons oder eines Feldes zu emulieren, abgeschlossen sein und damit eine Ausgabe erzeugen in der Zeit, die verfügbar ist zwischen Abtastpunkten. Auf diese Weise wird die Ausgabe für einen gegebenen Abtastpunkt berechnet und damit erzeugt vor der Annahme der Information für den nächsten Abtastpunkt. Diese Prozedur erlaubt einen Echtzeitbetrieb unter der Voraussetzung, dass, wenn die Berechnungszeit die Abtastperiode überschreitet, entweder zusätzliche Computer-Ressourcen in der Form eines zusätzlichen Prozessors und/oder Speichers bereit gestellt werden müssen, um die Berechnungszeit zu vermindern oder die Abtastfrequenz verringert werden muss, womit die Zeit für eine Berechnung erhöht wird. Die zunehmende Prozessleistung moderner Computer bedeutet, dass in der Praxis jegliche Beschränkung der Abtastfrequenz von keiner praktischer Bedeutung ist.

In Verbindung mit dem oben Gesagten wird es ebenso gewürdigt werden, dass die Berechnung, die von dem Computer ausgeführt werden soll, Rückkopplung einer Ausgabe beinhalten kann zu einer Abtastzeit, die dann die Eingabe für eine nachfolgende Berechnung zur nächsten Abtastzeit darstellt. Die nachfolgende Berechnung muss jedoch innerhalb der Abtastperiode abgeschlossen werden.

Fig. 8 veranschaulicht zwei verhältnismäßig einfache Felder. Die Eingabe- und Ausgabewellenformen von und zu diesen Feldern sind jeweils in Fig. 9 dargestellt. Man wird sehen, dass die Eingabe und die Ausgabe des ersten Feldes eine analoge Spannung umfassen von einem ECG. Für das andere Feld umfassen die vier Ausgaben die Ausgabe des spannungsgesteuerten Oszillators und die Ausgabe, wenn sie jeweils durch einen Tiefpassfilter, einen Hochpassfilter und einen Bandpassfilter geführt wird. Es sei bemerkt, dass, da diese Felder in keiner Weise miteinander in Beziehung stehen, obwohl all die Wellenformen gleichzeitig erzeugt und/oder dargestellt werden können, es für die Ergebnisse nicht notwendig ist, in irgendeiner Weise verbunden zu sein.

Bezug nehmend nun auf Fig. 10 veranschaulicht diese Zeichnung das Feld, das zuerst gezeichnet und dann kompiliert wird, um einen linearen Interpolations-  
 schaltkreis zu emulieren. Die konstante Spannungsquelle 45 wird angewandt als  
 5: Eingang auf drei spannungsgesteuerte Oszillatoren 46-48, die jeweils einen Sinus-  
 Wellenausgang, einen Quadrat-Wellenausgang und einen Rampenausgang auf-  
 weisen. Die Ausgabe der Sinuswelle wird auf 3 Hz eingestellt, während die Aus-  
 gabe der anderen zwei Generatoren 47 und 48 eingestellt wird auf die Abtastfre-  
 quenz von 20 Hz. Die Ausgabe des Sinus-Wellenoszillators 46 wird verwendet  
 10: um jede der drei Anzeigen 50-52 anzusteuern. Die Ausgabe des Quadrat-  
 Wellenoszillators 47 wird verwendet, um zwei Abtast- und Halteschaltkreise 53,  
 54 anzusteuern, die getrennt sind von einem Zeitverzögerungsschaltkreis 55, der  
 eine Verzögerung aufweist, die im Wesentlichen gleich ist der Periode zwischen  
 dem Abtasten. Dies stellt sicher, dass die Ausgaben der zwei Abtast- und Halte-  
 15: schaltkreise 53, 54 die Ergebnisse aufeinander folgender Abtastungen darstellen.

Eine weitere Spannungsreferenz 56 wird eingestellt auf ein Volt und umfasst eine  
 Eingabe an einen Verminderer 57. Die lineare Interpolation wird ausgeführt durch  
 die zwei Multiplizierer 58, 59 und den Addierer 60. Die Multiplizierer multipli-  
 20: zieren den Rampen-Gradienten mit dem richtigen Verhältnis, das bestimmt wird  
 durch die Amplitudendifferenzen von aufeinander folgenden Abtastungen, welche  
 in den Schaltkreisen 53, 54 gehalten werden.

Die „Eingabe“-Sinuswelle bei 3 Hz, welche die Bildschirmanzeige 50 darstellt, ist  
 25: in Fig. 11 veranschaulicht, ebenso wie die abgetastete Sinuswelle, die die Anzeige  
 51 darstellt. Die lineare Interpolation, welche erzeugt wird von der abgetasteten  
 Sinuswelle stellt die Anzeige 52 dar und ist ebenso veranschaulicht in Fig. 11.

Fig. 12 veranschaulicht ein Feld, welches eine phasenverriegelte Schleife bildet. Eine Spannungsreferenz 61 stellt wiederum die Eingabe eines spannungsgesteuerten Sinuswellenoszillators 62 dar, wobei die Referenzspannung von 61 so gesteuert ist, dass die Frequenz, die vom Oszillator 62 erzeugt wird, eingestellt ist auf 51 Hz. Die Ausgabe des Oszillators 62 wird benutzt, um einen Pulsschaltkreis 63 anzusteuern. Die Ausgabe des Pulsschaltkreises 63 steuert ihrerseits drei Anzeigen 64-66 und bildet das Anzeigesignal für die Anzeige 64.

Zusätzlich wird die Ausgabe des Pulsschaltkreises 63 verwendet, um einen Abtast- und Halteschaltkreis 67 anzusteuern, der einen Eingang aufweist, der gebildet wird vom Ausgang des spannungsgesteuerten Sinuswellenoszillators 68, der so eingestellt ist, dass er eine Mittelfrequenz von 50 Hz aufweist. Die Ausgabe des Spannungssteuerungsozillators 68 bildet die Ausgabe des Abtast- und Halteschaltkreises 67, der im Wesentlichen die Phasendifferenz darstellt zwischen den Signalen von den Oszillatoren 68 und dem Pulsschaltkreis 63. Dieses Fehlersignal wird angezeigt durch die Anzeige 66, wobei die Ausgabe des Spannungssteuerungsozillators 68 angezeigt wird von der Anzeige 65. Die drei Anzeigen 64, 65 und 66 von Fig. 12 sind jeweils in Fig. 13 erläutert.

Ein Feld zur Lösung einer Differentialgleichung zweiter Ordnung ist veranschaulicht in Fig. 14. Hier werden drei Rückkopplungsschleifen FB1 - FB3 bereit gestellt. Um die Anfangsbedingungen einzustellen, wird eine Spannungsreferenz 70, die auf ein Volt eingestellt ist, verwendet, um eine Eingabe bereit zu stellen an einen Abtast- und Halteschaltkreis 71 und die invertierende Eingabe eines Komparators 72. Die Ausgabe des Abtast- und Halteschaltkreises 71 wird zurückgeführt zum Komparator 72 und ebenso zu einem der beiden Multiplizierer 73, 74. Die Ausgabe der Multiplizierer 73, 74 wird zusammengezählt im Addierer 75 und multipliziert mit der Verstärkungsleistung des Verstärkers 76, bevor sie integriert wird durch den ersten von zwei Integraloren 77, 78. Ein weiterer Verstärker 79 und Multiplizierer 80 vervollständigen den Schaltkreis. Die Ausgaben jedes der

Integratoren 77 und 78 bilden die Wellenformen, die von den beiden Anzeigen 81 und 82 angezeigt werden, welche beide angesteuert werden von der Ausgabe des ersten Integrators 77.

- 5 Die Ergebnisse sind veranschaulicht in Fig. 15 und zeigen, dass sowohl die Anfangslösung, die den Anfangsbedingungen entspricht, als auch die Ausgabe, welche die Lösung der Differentialgleichung darstellt, endlos läuft ohne ersichtliche Verluste oder Gewinne innerhalb der Begrenzung von Quantisierungsfehlern. In einer Realisierung durch einen tatsächlichen elektronischen Schaltkreis, welche  
10 die Wirkung der Lösung der Differentialgleichung zweiter Ordnung hervorbringen soll, würde die Verwendung tatsächlicher Kondensatoren in Verlust oder fortschreitenden Abfall oder Anstieg des Ausgabesignals münden. Mit der oben beschriebenen Anordnung gibt es jedoch, weil die Lösung fortwährend berechnet wird, weder einen ersichtlichen Abfall noch einen instabilen Anstieg, der zu einer  
15 Sättigung führt. Dies stellt eine wesentliche Verbesserung gegenüber Analogcomputern nach dem Stand der Technik dar.

- Fig. 16 ist ein Beispiel eines komplexeren Feldes, das kompiliert werden kann in Übereinstimmung mit der bevorzugten Ausführungsform der vorliegenden Erfindung. Es wird aus Fig. 16 ersichtlich werden, dass vier schnelle Fourier-  
20 Transformationen, sechs Differenzialsignalgeneratoren, numerische Anzeigen, Balkendiagramme, vier Referenzspannungen, vier 40 dB/Dekadenfilter, ein "Fläche unter der Kurve"-Berechner und verschiedene Komparaforen, Multiplizierer und Ähnliches alle bereitgestellt werden.

- 25 Weil die Ergebnisse, wie z.B. die Wellenformen 11, erläutert in Fig. 6, in Speicher gespeichert werden, ist der Benutzer in der Lage, diese gespeicherten Echtzeitergebnisse wieder abzuspielen. Darüber hinaus können verschiedene Abschnitte dieser Ergebnisse extrahiert werden für irgendeinen besonderen Zweck.

und selbst in die Vorrichtung zurückgeführt werden unter Verwendung eines "Von-Platte"-Icons. Dies erlaubt, dass die Signalverarbeitung so angepasst wird, dass ein gewünschtes Merkmal des Eingangssignals detektiert wird. Das gespeicherte Ergebnis kann ebenso fortwährend in Echtzeit wiedergegeben werden oder schneller oder langsamer, wie gewünscht. Ebenso können Anfangsbedingungen aufgestellt werden unter Verwendung des vorher gespeicherten Ergebnisses von einem ersten Feld als die Anfangseingabebedingung für ein zweites Feld.

Weiterhin können, weil die Ergebnisse gespeichert sind, die gespeicherten Daten editiert und exportiert werden auf Spreadsheets, graphische oder statistische Dienstprogramme, wie z.B. EXCEL, LOTUS 123 und Ähnliches. Dies erlaubt es, die graphischen Ergebnisse von Experimenten zu kombinieren mit Text, der die Natur der Experimente und die Natur der Schlussfolgerungen beschreibt.

Die Fenster, wie diejenigen, die in Fig. 8 angegeben sind, sind z.B. ebenso kompatibel mit Multimedia-Standards für den IBM PC. Die Spezifikation für den Fenster-Icon kann eingestellt werden, "PAL" zu sein, welches ein herkömmliches Signal von einer Videokamera empfängt, konvertiert und dann das Videosignal auf dem Bildschirm anzeigt. Falls es gewünscht wird, kann dieses Videosignal geframegrapped werden zu Videodaten, die gespeichert werden können im Computerspeicher. Falls gewünscht, kann das herkömmliche Videokamerasignal auch gleichzeitig auf einem VCR gespeichert werden. Die gespeicherten Videodaten können wieder abgespielt werden auf dieselbe Weise wie jede andere Wellenform, die von der Vorrichtung aufgenommen wurde. Auf diese Weise kann eine Videobildkorrelation mit anderen Signalwellenformen erreicht werden. Die Zeitmarke, welche auf das herkömmliche VCR-Videoband aufgebracht ist, wird in dieser Ausführungsform synchronisiert mit den Signaldaten, die von der Vorrichtung aufgenommen und darin gespeichert werden. Diese Anordnung ist besonders vorteilhaft, da sie erlaubt, elektrische Ergebnisse klar zu identifizieren mit der optischen Aufzeichnung der Ereignisse, welche die Ergebnisse erzeugt haben.



Die Bereitstellung von Icon-Spezifikationen bedeutet, dass die Vorrichtung besonders anpassungsfähig ist. Zum Beispiel kann ein Feld gebildet werden mit einer Ansicht zur Ausführung einer Analyse von z.B. den elektrischen Spannungen, die vom menschlichen Herzen erzeugt werden. Falls als Ergebnis der Analyse der Ergebnisse von diesen Experimenten gedacht ist, dass ein bestimmtes Arzneimittel eine bestimmte vorteilhafte Wirkung erzeugt, dann muss das Feld selbst nicht geändert werden, um es zu erlauben, dass dieselbe Messung ausgeführt werden kann mit einer Ratte, welcher das Arzneimittel verabreicht wurde. Alles, was erforderlich ist, geändert zu werden, ist die Spezifikation derjenigen Icons, die verwendet werden z.B. als eine Referenz. Zum Beispiel können das tatsächliche Spannungsniveau und die Frequenz angepasst werden, um den verschiedenen elektrischen Ausgängen Rechnung zu tragen (z.B. Spannungsniveaus und verschiedene Pulsraten) zwischen Menschen und Ratten.

Weiterhin bedeutet die Fähigkeit, Felder beliebig zu erzeugen, dass die Vorrichtung in der Lage ist, die Beziehung oder Unabhängigkeit zwischen Signalen zu messen und auf diese Weise eine weitere Dimension zu den Ergebnissen hinzuzufügen, welche erzielt werden können. Zum Beispiel kann eine Herzrate korreliert werden mit oder multipliziert werden mit einer Beatmungsrate eines Patienten, um ein drittes Signal zu ergeben, das betrachtet werden soll als Signatursignal, welches ein Ergebnis darstellt, welches dem Benutzer wichtig ist. Viele solche Signale, selbst von gemischten Einheiten, können auf diese Weise assoziiert werden, falls nötig geeignet skaliert werden und damit verglichen werden mit vorher im Speicher gespeicherten Daten.

Dem Fachmann wird es offenkundig erscheinen, dass in dem oben beschriebenen System alle Funktionen integriert sind, die erforderlich sind für eine analoge Workstation. Bis zum heutigen Tag gab es noch kein vollständig integriertes Sys-

tem. Stattdessen gab es eine Vielfalt von partiellen Funktionen. Insbesondere stellen die folgenden Merkmale Komponententeile des integrierten Ganzen des wissenschaftlichen Emulators der bevorzugten Ausführungsform dar:

- 1 Ein Mehrzweck-/Mehrfunktions-Analogmodul, das eingibt/ausgibt digitale und analoge Signale und andere Funktionen.
- 2 Ein Echtzeit-Video, gleichzeitig sowohl in PAL oder Bildform- und Wellenformanzeigen, in bis zu 40 Fenstern auf einem Bildschirm sind verfügbar. Die oszillographischen Anzeigen (eine Art von Anzeige-Verfahrensweise) können Abtastraten in der Größenordnung von 25 KHz oder besser anzeigen.
- 3 Die Signalverarbeitung wird nicht gesteuert durch eine sequentielle syntaktische "Leitungssprache" im Prosastil (von Neumann), sondern durch einen einfach verstandenen und sofort verwendeten parallelen graphischen Compiler, der von Nicht-Programmierern verwendet werden kann.
- 4 Wellenformen und graphische Felder in Bildform werden sofort zu bestehenden PC- Werkzeugen transportiert, wie z.B. Wortprozessoren, Spreadsheets und "Offline"-Analysesoftware für die Berichterzeugung/Aufzeichnungen, etc.
- 5 Auf nicht-flüchtigen Speicher kann derart zugegriffen werden, dass er als "Quelle" von Daten dient, die zurückgeführt werden sollen in ein Feld für andere Ergebnisse. Ebenso ist das Korrelieren eines Wellenformmusters, das auf der Platte gespeichert ist, mit einer beliebigen ankommenden Wellenform möglich, um eine Gestaltdetektion zu erreichen. (Template matching).

6. Netzwerkkompatibilität. Das System erlaubt es, dass seine verschiedenen Funktionen über ein Netzwerk verteilt sind. Das heißt, eine Speicherung kann ausgeführt werden in einem PC in einem Büro, oder Wellenformdaten können gespeichert werden in einem PC, um auf einem anderen angezeigt zu werden. Ebenso können Programme und Daten über das Netzwerk transportiert werden zu einem anderen ähnlichen wissenschaftlichen Instrumentenemulator zur Analyse. Die Daten und Programme sind verbunden oder "gebündelt" zur Netzwerkübertragung.
- 10 In der bevorzugten Ausführungsform nimmt das System die Form einer Signalverarbeitungsvorrichtung an, die eine Vielzahl von proprietärer Hardware und Software umfasst, die in einem IBM PC enthalten ist mit Datenerfassungsverstärkern, welche in Floppy-Disk-Schächten untergebracht sind und Prozessor-PCBs und Echtzeit-Videokarten, die aufgesteckt sind auf dem PC-Systembus. Die Vorrichtung in einer Konfiguration enthält zusätzlich 2 Mehrzweckprozessoren und
- 15 weiter 2 Signalprozessoren. Diese kombinierten Prozessoren werden programmiert durch die Verwendung eines graphischen Compilers auf dem VGA-Bildschirm auf eine solche Weise, dass der PC-Host (betrieben bei etwa 3.000.000 Instruktionen pro Sekunde) das residente Vorrichtungssystem (betrieben bei etwa fünfzehn Millionen Instruktionen pro Sekunde) als eine Erweiterung seines
- 20 Speichers ansieht. All die anderen Prozessoren sehen einander auch als Erweiterungen ihres eigenen Speichers. Auf diese Weise kann die Multitasking-Software mehrfache Prozessoren verwalten, wobei jeder Task auf den verschiedenen Prozessoren mit dem anderen Task/den anderen Tasks kommuniziert und/oder mit
- 25 Prozessoren) durch Weitergeben von Zeigern in den gemeinsamen Speicher. Auf diese Weise wird keine besondere Kommunikationshardware oder software erforderlich. Der graphische Compiler, der ausführbaren Objektcode für das installierte Subsystem kompiliert, erlaubt den strengen Gebrauch von Mathematik, die so schnell verarbeitet wird, dass sie Echtzeit simuliert, wobei alle notwendigen
- 30 Verarbeitungsschritte innerhalb einer einzigen Abtastzeit durchgeführt werden.

Zusätzlich zu der funktionalen Zulänglichkeit des Programmierungssystems des graphischen Compilers ist das Programmsystem, das Icons benutzt, auf eine solche Weise aufgebaut, dass kompilierte Icons selbst zusammengesetzt sein können aus einfachen Icons. Dies erlaubt, dass ein mathematischer Aufbau angewandt werden kann auf zwei Arten von Datenströmen, die bei der Verarbeitung vorkommen. Einer dieser Typen von Datenströmen ist der "erfasste Datenstrom" von dem Analog-zu-Digital-Umwandlungsprozess und der andere Typ ist ein "Ansteuerungsstrom". Ein Ansteuerungsstrom ist ein Signal, das erfasst wird von dieser Zwischenumwandlung und mit dem andere Signale, von denen gewünscht wird, dass sie angezeigt werden in einer Zeitbeziehung, oder es wird erzeugt von einer Zeitbasis, die in dem Feld aufgestellt ist. Am häufigsten wird der Ansteuerungsstrom erzeugt durch eine Verarbeitungsfunktion, wobei die Verarbeitung auf einem oder mehreren Kanälen ein Signal erzeugt, das Anzeigen ansteuern soll, eine Verarbeitung anzusteuern, wie z.B. eine Mittelwertbildung oder eine Speicherung anzusteuern. Im Allgemeinen ist es die komplexe Ansteuerungsfähigkeit des Systems, die vereinfacht durch den graphischen Compiler dargestellt ist, welcher es erlaubt, dass Parameter akkumuliert werden, welche verschiedene Wellenformen oder Epochen von Wellenformen charakterisieren. Dies erlaubt deren Auswahl durch Experten, um einen "Beispielsatz" zu begründen.

### INDUSTRIELLE ANWENDUNG

Die Kostengünstigkeit der Vorrichtung der bevorzugten Ausführungsform kann aus dem Folgenden in Betracht gezogen werden. Die Vorrichtung kann als irgendeine der folgenden Vorrichtungen oder als irgendeine Kombination der folgenden Vorrichtungen (oder Vielfachen davon), die zusammen in einem Feld angeordnet sind, arbeiten:

	VORRICHTUNG	KOSTEN A\$
1.	4-Kanaldiagramm-Recorder (500 Hz/Kanal)	10.000
2.	FFT-Analysierer (1 Hz-20 KHz; max 512 Punkte; 200 Analysen/sec.)	13.000
3.	Frequenzzähler (0,01% Genauigkeit; 0,01 Hz-10 MHz)	1.000
5.	4-Funktionsgenerator (Sinus-, Quadrat-, Rampen- und Dreieck-Wellenformen; 0,01 Hz - 2 MHz)	5.000
5.	Datenaufzeichnung (PC-basiert)	4.000
6.	Leistungsüberwacher (Volt x Spannung; isolierte Eingänge)	4.000
7.	2-Kanalübergangsrecorder	3.500
10	8-Elektrophysiologischer Überwacher (EOG, EEG, EMG, ERP)	20.000

Die obigen Ausrüstungsteile ergeben Kosten von etwa 60.500 A\$, wohingegen der Wiederverkaufspreis eines wissenschaftlichen Instrumentenemulators gemäß der bevorzugten Ausführungsform, welcher in der Lage ist, die obigen Funktionen zu emulieren, in der Nähe von 15.000 A\$ liegt (unter der Annahme, dass der Benutzer einen geeigneten Personal Computer besitzt, der sowohl die notwendige Hardware als auch die notwendige Software ablaufen lassen kann).

Weiterhin ist die große Anzahl paralleler Eingänge in dem integrierten System gut geeignet für das Studium und die Analyse paralleler Systeme, wie z.B. von Ökosystemen, Biosystemen, Maschinensystemen, etc.

Das Vorangegangene beschreibt lediglich eine Ausführungsform der vorliegenden Erfindung und Modifikationen, die dem Fachmann offensichtlich sind, können daran vorgenommen werden, ohne vom Schutzzumfang der vorliegenden Erfindung abzuweichen.

COPYRIGHT HINWEIS:

Die Programmlisten, welche in den Anhängen I-IV enthalten sind, unterliegen dem Copyright, welches vom Anmelder gehalten wird, und sie dürfen nicht reproduziert werden in jeglicher Form ohne die ausdrückliche vorherige schriftliche Erlaubnis des Anmelders.



05.03.02

- 31 -

```
#####
### PULSE Sheet WINDOW Definition
#####
*/
.....
PULSE sheet Variables
.....
*/
/* Local string Arrays.
char title[] = "PULSE GENERATOR SHEET";

/* ##### Sheet Title Definition ##### */
TITLE pulse_sheet_title =
{
    NULL,
    -1,
    BLACK, WHITE,
    title,
    NULL,
};

/* ##### Sheet Control Definition ##### */
CONTROL pulse_sheet_control =
{
    NULL,
    -1,
    IWHITE, GREY,
    LOGO,
    0.0,
    0, NULL,
    0.0, 0.0,
    0.0, 0.0,
};

/* ##### Edit Region Structure Definitions. ##### */
/*
int check_duration(int);
EDIT_DEFN pulse_duration =
{
    UNSIGNED_INTEGER,
    5,
    BLACK, WHITE, IWHITE, GREY,
    0.0,
};
```



000000

- 3 -

```
"Pulse ON State Duration (Samples):"  
&pulse_vars.pa.duration  
check_duration  
NULL  
);
```

```
int check_on_level(int)  
EDIT_DEFN pulse_on_level =
```

```
{  
    INTEGER  
    6  
    BLACK,WHITE,WHITE,GREY  
    0.0  
    "Pulse State Levels (mV) - ON:"  
    &pulse_vars.pa.on_level  
    check_on_level  
    NULL  
};
```

```
int check_off_level(int)  
EDIT_DEFN pulse_off_level =
```

```
{  
    INTEGER  
    6  
    BLACK,WHITE,WHITE,GREY  
    0.0  
    "OFF:"  
    &pulse_vars.pa.off_level  
    check_off_level  
    NULL  
};
```

```
int check_trigger_level(int)  
EDIT_DEFN pulse_trigger_level =
```

```
{  
    INTEGER  
    6  
    BLACK,WHITE,WHITE,GREY  
    0.0  
    "Trigger Level (mV):"  
    &pulse_vars.pa.trigger_level  
    check_trigger_level  
    NULL  
};
```

000000

```
int check_edge(int);
EDIT_DEFN pulse_trigger_edge =
{
    CHAR,
    BLACK, WHITE, IWHITE, GREY,
    0.0,
    "Edge (+,-): ",
    &pulse_vars.pa.edge,
    check_edge,
    NULL,
};
```

```
int check_slew(int);
EDIT_DEFN pulse_trigger_slew =
{
    UNSIGNED_INTEGER,
    5,
    BLACK, WHITE, IWHITE, GREY,
    0.0,
    "Slew (mV/mS): ",
    &pulse_vars.pa.slew,
    check_slew,
    NULL,
};
```

```
int check_mode(int);
EDIT_DEFN pulse_trigger_mode =
{
    CHAR,
    MAX_TRIG_MODE,
    BLACK, WHITE, IWHITE, GREY,
    0.0,
    "Mode (SINGLE SWEEP, NORMAL): ",
    &pulse_vars.pa.trigger_mode,
    check_mode,
    NULL,
};
```

```
/* ##### Sheet Region Definitions ##### */
SCREEN_REGION pulse_sheet_regions[] =
{
    EDIT_CLASS,
```

08.03.02

= 34 =

```
4.2.0.0;  
&pulse_duration;
```

```
EDIT_CLASS;  
4.5.0.0;  
&pulse_on_level;
```

```
EDIT_CLASS;  
28.7.0.0;  
pulse_off_level;
```

```
EDIT_CLASS;  
4.10.0.0;  
&pulse_trigger_level;
```

```
EDIT_CLASS;  
12.12.0.0;  
&pulse_trigger_edge;
```

```
EDIT_CLASS;  
12.14.0.0;  
&pulse_trigger_slew;
```

```
EDIT_CLASS;  
4.17.0.0;  
&pulse_trigger_mode;  
};
```

```
#define NR_REGIONS (sizeof(pulse_sheet_regions)/  
sizeof(SCREEN_REGION))
```

```
/* ##### PULSE Sheet Window Definition ##### */
```

```
AMWINDOW pulse_sheet_window[] =  
{  
0.0.0.0,  
(VIRTUAL_MAX_X/2), (VIRTUAL_MAX_Y/2),  
WINDOW_WIDTH, WINDOW_HEIGHT,  
0.0,  
BLACK_WHITE,  
NULL,  
NULL,  
NULL,  
&pulse_sheet_title,
```

09.03.02

35

```
&pulse_sheet_control;  
0.NULL;  
NR_REGIONS.pulse_sheet_regions  
};
```

```
/* ##### Error Messages: ##### */  
ERROR_MSG pulse_sheet_errors[] = {  
0 /*
```

RANGE\_ERROR,

"Pulse Duration Error\n\nThe Pulse\  
Duration specified is Invalid\n\nValid range is from 1 to 65535 samples"

1 /\*  
RANGE\_ERROR,

"ON State Level Error\n\nThe ON state\  
level specified for the pulse is Invalid\n\nValid Range is -10000 to 10000  
mV"

2 /\*  
RANGE\_ERROR,

"OFF State Level Error\n\nThe OFF state\  
level specified for the pulse is Invalid\n\nValid Range is -10000 to 10000  
mV"

3 /\*  
RANGE\_ERROR,

"Trigger Level Error\n\nThe Trigger\  
level specified is Invalid\n\nValid Range is -10000 to 10000 mV"

4 /\*  
RANGE\_ERROR,

"Edge Error\n\nThe Trigger Edge\  
specified is Invalid\n\nValid Edges are + or -"

5 /\*  
RANGE\_ERROR,

"Trigger Mode Error\n\nThe Mode\  
specified is Invalid\n\nValid Modes are SINGLE SWEEP or NORMAL"

6 /\*  
RANGE\_ERROR,

"Slew Rate Error\n\nThe Slew\  
rate specified is Invalid\n\nValid Range is 0 to 10000 mV/mS"

05 05 12

-36-

```
/*
*****
**** PULSE Sheet Definition ****
*****
*/
```

```
SHEET pulse_sheet =
{
    pulse_sheet_window,
    &pulse_vars.ip,
    pulse_sheet_errors
};
```

```
/* Local Save Area of Global Variable Pointer */
GLOBAL_VARS *global_variables;
```

```
*****
PULSE Sheet Handler
*****
```

```
/*
SHEET *pulse_sheet_handler(GLOBAL_VARS *global)
```

```
/* preset the global variables static */
global_variables = global;
```

```
return( sheet_handler( global, USER_X_SIZE, USER_Y_SIZE,
    NR_REGIONS, &pulse_sheet, pulse_sheet_regions ));
```

```
*****
Check Duration (0)
*****
```

```
int check_duration(int checkpoint)
```

```
{
    return( check_range( checkpoint, &pulse_vars.pa.duration,
        MAX_UNSIGNED, 1, 0 ));
```

```
etc...
```

00-10-00

- 37 -

## ANHANG II

### Compiler-Elemente

Der folgende Code stellt ein Beispiel dar des Compiler-Elements, welches für einen gesteuerten Pulsgenerator verwendet wird.

```

.....
/*
.....
                                Pulsegen Icon Compiler Overlay
.....
*/

#include <stdio.h>
#include <stdlib.h>
#include "compiler.h"
#include "library.h"
#include "comms86.h"
#include "pulsegen.h"
.....
                                Local Definitions
.....
*/

void init_trigger_params(PULSE_ICON_PARAM *);

/*
.....
                                Pulse Process
.....
*/

#define NR_PULSE_INPUTS 1
#define NR_PULSE_OUTPUTS 1

/* I/O Buffer Lists */
unsigned int pulse_input_ids[NR_PULSE_INPUTS] = {1};
unsigned int pulse_output_ids[NR_PULSE_OUTPUTS] = {0};

/* Cell Parameters */
struct PULSE_struct pulse_param;

/* Processes in this Icon */
struct proc_type pulse_proc =

    NULL /* the next process defined */
    PULSE /* the cell required */
    pulse_input_ids /* array of input buffer numbers */
    pulse_output_ids /* array of output buffer numbers */
    &pulse_param /* array of the initial values of the cells parameters */

```

08-03-02

= 38 =

```

0      /* the base xdran address of variables for this cell */
1

/*
#####
### Trigger Process
#####
*/
#define NR_TRIGGER_INPUTS 1
#define NR_TRIGGER_OUTPUTS 1

/* Trigger input ID specifications */
unsigned int trig_input_ids[NR_TRIGGER_INPUTS] = {0};

/* Trigger output ID specifications */
unsigned int trig_output_ids[NR_TRIGGER_OUTPUTS] = {1};

/* Trigger parameters */
struct THRESH_struct trig_params;

/* Process Descriptor */
struct proc_type trigger_proc =
{
    &pulse_proc, /* the next process defined */
    THRESH, /* the cell id required */
    trig_input_ids, /* array of input buffer numbers */
    trig_output_ids, /* array of output buffer numbers */
    &trig_params, /* array of the initial values of the cells
parameters */
    0 /* the base xdran address of variables for this
cell */
};

/*
#####
### PULSEGEN CONFIGURATION
#####
*/
unsigned int int_interconnect_ids[] =
{
    &trig_output_ids[0],
    &pulse_input_ids[0]
};

```

00-00-00

```

#define NR_INTERCONNECT_IDS (sizeof(int_interconnect_ids)/
                             sizeof(unsigned int*))

unsigned int *ext_input_ids[] =
{
    &trig_input_ids[0]
};

#define NR_EXT_INPUT_IDS (sizeof(ext_input_ids)/
                          sizeof(unsigned int*))

unsigned int *ext_output_ids[] =
{
    &pulse_output_ids[0]
};

#define NR_EXT_OUTPUT_IDS (sizeof(ext_output_ids)/
                           sizeof(unsigned int*))

/* PULSEGEN Dsp System Configuration */
ICON_CONFIG pulse_config =
{
    2, /* nr_processes */
    &trigger_proc, /* process list */
    0, /* nr_raw */
    NULL, /* raw list */
    0, /* nr_screen */
    NULL, /* screen list */
    0, /* nr_virtual */
    NULL, /* virtual list */
    NR_INTERCONNECT_IDS, /* number of int interconnect ID's */
    int_interconnect_ids,
    NR_EXT_INPUT_IDS, /* number of external input ID's */
    ext_input_ids,
    NR_EXT_OUTPUT_IDS, /* number of external output ID's */
    ext_output_ids,
    NULL, /* Overlay Struct */
    NULL, /* Icon parameter key */
    NULL, /* next_proc */
    NULL, /* next_raw */
    NULL, /* next_scrn */
    NULL, /* next_virt */
    NULL /* next_icon */
};
    
```



06-03-02

- 40 -

```

/*****
 *
 * Request Block Size
 *****/
*/
unsigned int mem_block_size = 0;

/*****
 *
 * Pulsegen Descriptor Handler
 *****/
*/
void *pulsegen(GLOBAL_VARS *gv, void *mem_block, ICON_PARAM
*ip)
{
    PULSE_ICON_PARAM *pip;

    /* Check for memory block request */
    if (mem_block == NULL)
        return(&mem_block_size);

    /* Index pulse icon parameters */
    pip = (PULSE_ICON_PARAM *) &ip[1];

    /* Set the pulse state levels */
    pulse_param.WIDTHPU = pip->duration;
    pulse_param.HIGHPU = calc_AD_val(pip->on_level);
    pulse_param.LOW_PU = calc_AD_val(pip->off_level);

    /* Set the Trigger Cell Variables */
    init_trigger_params(pip);

    /* Return Configuration Pointer */
    return(&pulse_config);
}

/*****
 *
 * Init Trigger Params
 *****/
*/
void init_trigger_params(PULSE_ICON_PARAM *pip)
{
    /* Transfer the level parameter */
    trig_params.THRETH = calc_AD_val(pip->trigger_level);

    /* Preset trigger width */
}

```

15.03.12

43

```
trig_params.WINSTH = 1;
```

```
/* Transfer the Slew Parameter */
```

```
trig_params.SLEWTH = pip->slew;
```

```
/* Preset Dead Period */
```

```
trig_params.DEADTH = 0;
```

```
/* Transfer the Trigger Edge Setting */
```

```
if (pip->edge[0] == POSITIVE_EDGE)
```

```
    trig_params.POSTTH = FLAG_POS_EDGE;
```

```
else
```

```
    trig_params.POSTTH = FLAG_NEG_EDGE;
```

```
/* Transfer the One Shot Status */
```

```
if (strcmp(pip->trigger_mode, SINGLE_SWEEP_TRIGGER) == 0)
```

```
    trig_params.ONESTH = FLAG_ONE_SHOT;
```

```
else
```

```
    trig_params.ONESTH = FLAG_CONTINUOUS;
```

```
}
```

05.03.02

-42-

### ANHANG III

#### Zellen-Code

Der folgende Code wird verwendet, um den Pulsgenerator auf dem DSP-Chip zu implementieren. Er ist geschrieben in TMS320 Assembler-Code.

Cell :- PULSE

#### Synopsis

- This cell generates a pulse once every time a trigger occurs.
- It has one input which is a trigger stream.
- It has one output which is the pulse train produced.

The cell has the following functioning:-

read in the trigger buffer

for each point in the buffer:-

if waiting for the trigger to go high:-

if trigger is high:-

change state to 1.  
reset pulse width counter.

else if trigger is low

change to state 0

if count > 0:

count--  
output = high

else output = low

finish the cell

IDT PULSE  
COPY WHERELIB

# SYSTEM VARIABLES

REF	RG1	WORKING REGISTERS
REF	RG2	
REF	RG3	
REF	ONE	THE NUMBER ONE
REF	BUFSIZ	THE SIZE OF BUFFERS
REF	I	A COMMONLY USED LOOP COUNTER
REF	FRESH	THE ALLOC BASE PTR
REF	ZERO	ZERO

# SYSTEM CALLS

REF	ENDCEL	THE END OF THIS CELL
-----	--------	----------------------

# LIBRARY ROUTINES

REF	READCH	READ IN A CHANNEL
-----	--------	-------------------

# THIS CELLS VARIABLES

REF	TRIGPU	INPUT. THE CELLS TRIGGER INPUT.
REF	OUTSPU	OUTPUT. THE CELLS OUTPUT.
REF	BUFFPU	TEMP ARRAY. USED TO BUFFER POINTS.
REF	STATPU	STATIC. STATE OF TRIGGER DETECTION.
REF	WIDTPU	PARAM. WIDTH OF PULSE

00-03-02

44

REF HIGHPU  
REF LOWSPU  
REF COUNPU

PARAM HIGH PULSE LEVEL  
PARAM LOW PULSE LEVEL  
STATIC THE NUMBER OF POINTS  
LEFT IN THE PULSE

# CONSTANTS

## COPY CASPDEF

PULSE  
INPUTS  
TRIGPU

OUTPUTS  
OUTSPU

## DEF PULSE

PULSE

read in the trigger buffer

LARK ARI,BUFFPU  
LAC TRIGPU  
SACL RGI  
CALL READCH

for each point in the buffer:

for (i = 0; i < buf\_size; i++)

LAR ARO,BUFSEZ  
LARK ARI,BUFFPU  
LAC OUTSPU  
ADD ONE,1  
SACL RGI  
SXRAM RGI

LINE UP WITH DATA

000002

PULSFI      BANZ PULSCI

B      ENDCEL  
PULSCI

if trigger is high:-

READ IN THE TRIGGER

LARP ARI  
LAC \*0,ARO  
BZ      ELSE1

if waiting for the trigger to go high:-

LAC STATPU  
BNZ DOOUT

change state to 1.  
reset pulse width counter.

LAC ONE  
SACL STATPU  
LAC WIDTPU  
SACL COUNPU  
B      DOOUT

else change to state 0

ELSE1ZAC  
SACL STATPU

if (count > 0)

DOOUT      LAC COUNPU  
BLEZ ELSE2

count--  
output = high

SUB ONE  
SACL COUNPU

000000

46

WXDRAM HIGHPU  
B PULSF1

else output = low

NOP  
ELSE2 WXDRAM LOWSPU  
B PULSF1

finish the cell

END

## ANHANG IV

### DSP-System-Code:

Der folgende Code implementiert das System, welches von DSPs in einer AMLAB-Umgebung verwendet wird:

### MODULE SYSTEM

#### SYNOPSIS

- DUE TO THE PERFORMANCE REQUIREMENTS OF THE TMS320 IT
- IS NECESSARY TO IMPLEMENT EACH PROCESSING ELEMENT AS
- A CELL IN A INTERPRETER TYPE SITUATION. AN EXPLANATION
- OF THIS INTERPRETER FOLLOWS. NO ATTEMPT TO EXPLAIN
- WHY THINGS ARE DONE THE WAY THEY ARE IS GIVEN HERE AS
- THESE CAN BE FOUND IN THE ACCOMPANYING
- SPECIFICATIONS.

#### DEFINITION OF TERMS

- **CELL** A PROCESSING ELEMENT. THE TOTAL PROCESSING TO BE DONE IS BROKEN DOWN INTO MODULES THAT ARE EASY TO IMPLEMENT.
- **EVENT** AN EVENT IS SOMETHING THAT HAPPENS. IT MAY BE INTERNAL TO THE SYSTEM SUCH AS A BEAT IS PROCESSED TO A CERTAIN LEVEL OR EXTERNAL SUCH AS AN INTERRUPT. THIS EXECUTIVE IS AN EVENT DRIVEN ONE. IN THAT IT IS EVENTS WHICH ULTIMATELY DETERMINE WHICH SEQUENCES ARE RUN AND IN WHAT ORDER.
- **SEQUENCE** A PRE-DETERMINED ORDER IN WHICH CELLS ARE RUN.
- **ACTIVE** THE ACTIVE CELL IS THE CELL CURRENTLY BEING EXECUTED OR ABOUT TO BE EXECUTED.
- **CCB** A CELL CONTROL BLOCK. THIS CONTAINS ALL THE



04-03-02

- 48 -

## RELEVANT INFORMATION ABOUT A CELL NEEDED BY THE SYSTEM

- ENDSCEL THE ROUTINE WHICH TIDIES UP AFTER A CELL RELINQUISHES CONTROL, FINDS THE NEXT CELL IN THE SEQUENCE, AND INSTALLS THAT CELL

## DESCRIPTION OF THE SYSTEM

- THE SYSTEM COULD IN BROAD TERMS BE DESCRIBED AS AN INTERPRETER. CELL 0 IS THE HIGHEST LEVEL OF PROCESSING IN THE SYSTEM. IT DETERMINES WHICH SEQUENCES WILL BE RUN BASED ON THE INFORMATION CONTAINED IN THE STATUS. SEQUENCES IN TURN ARE COMPOSED OF CELLS. CELL 0 RUNS A SEQUENCE. THE SYSTEM TAKES THAT SEQUENCE AND RUNS EACH CELL IN TURN. INDIVIDUAL CELLS CAN CHANGE THE STATUS TO INDICATE TO CELL 0 WHAT IS THE RESULTS OF ITS PROCESSING. FURTHER MORE CELLS ARE ABLE TO ABORT THE CURRENT SEQUENCE WHICH RETURNS CONTROL TO THE CELL 0. CELL 0 IS DIFFERENT TO THE OTHER CELLS IN THAT IT ALWAYS SAVES A RETURN ADDRESS BEFORE RELINQUISHING CONTROL. THIS IS BECAUSE CELL 0 IS NEVER COMPLETED, RELINQUISHING CONTROL FREQUENTLY, AND RESTARTING WHERE IT LEFT OFF THIS ALLOWS US TO IMPLEMENT A MACRO LEVEL PROGRAM IN CELL 0, COMPOSED OF SEQUENCES WHICH IN TURN ARE COMPOSED OF CELLS.
- SEQUENCES ARE REFERENCED BY THE SEQUENCE REFERENCE LIST, WHICH, FOR EASE OF PROGRAMMING EXISTS AT THE BOTTOM OF EXTERNAL DATA RAM. EACH MEMBER IN THIS LIST POINTS TO A SEQUENCE LIST. EACH SEQUENCE LIST MEMBER IS A POINTER TO CELL CONTROL BLOCK. EVERY SEQUENCE HAS A SEQUENCE LIST. EACH CONSECUTIVE CELL IN A SEQUENCE IS POINTED TO BY A CONSECUTIVE MEMBER IN THE SEQUENCE LIST.
- A CELL IS RESTORED BY LOADING INTERNAL DATA MEMORY WITH THE CELLS STATIC VARIABLES. THESE ARE POINTED TO BY A MEMBER OF THE CCB. THE CELLS START ADDRESS IS ACCESSED BY USING THE CELLID. ANOTHER MEMBER OF THE CCB AS AN OFFSET TO A LOOK-UP TABLE STORED IN PROGRAM MEMORY CALLED THE CSASB. A CELL IS SAVED BY THE REVERSE PROCESS. THE FINAL ELEMENT OF A CCB IS A CONSTANT. A CONSTANT IS A STATIC VARIABLE THAT DOES

NOT CHANGE ITS VALUE AND HENCE DOES NOT NEED TO BE  
STORED AWAY.

IDENT SYSTEM  
COPY WHERELIB

### STATIC VARIABLES USED BY THE SYSTEM

REF	CCBSPT	A POINTER TO THE CURRENTLY ACTIVE CCB
REF	CURSSQ	A POINTER TO THE CURRENTLY ACTIVE SEQUENCE
REF	ONE	
REF	TABLE	A POINTER TO THE VARIABLES TABLE IN IORAM
REF	RG1 RG2	
REF	CSASE	THE CELL ADDRESS BLOCK
DEF	ABORT	LET OUTSIDE TASKS KNOW ABOUT THE ROUTINES
DEF	ABORT1	
DEF	ENDCEL	
DEF	RUN	

### CONSTANTS USED BY THE SYSTEM

REF	CELVAR	THE START OF A CELL'S VARIABLES IN INTERNAL RAM
REF	COCCEB	THE ADDRESS OF CELL0 CCB

THE SYSTEM ENTRY POINT. THIS IS ONLY USED ON START

UP

05-03-02

-SD-

\*\*\*\*\*  
DEF SYSTEM  
PSEG  
SYSTEM B ABORT: SYSTEM ENTRY POINT  
\*\*\*\*\*

\*\*\*\*\*  
SAVE THE ENVIROMENT OF THE CURRENT CELL  
\*\*\*\*\*

SAVE RXDRAM RG2.CCB\$PT  
RXDRAM TABLE

RXDRAM RG1

RXDRAM RG1

LAR AR0RG1

LAR AR1CELVAR

SXGRAM TABLE

LARP 0

BANZ SAVELP

B SAVRET

SAVELP

LARP 1

WXDRAM \*+AR0

BANZ SAVELP

SAVRET

RET  
\*\*\*\*\*

READ IN THE CELL ID  
READ IN THE CURRENT  
CELLS VARIABLE TABLE  
READ IN THE NUMBER OF  
INSTALLED VARIABLES  
READ IN THE NUMBER OF  
STORED VARIABLES.

SET UP THE POINTER TO  
THE VARIABLES

ARP = 0  
IF THERE ARE ANY MORE  
VARIABLES SAVE THEM

WRITE OUT THE NEXT  
VARIABLE  
ARE THERE ANY MORE

\*\*\*\*\*  
RESTORE THE ENVIROMENT OF THE CURRENT CELL  
\*\*\*\*\*

00-13-02

= 9 =

RESTOR

RXDRAM RG2CCBSPT  
RXDRAM TABLE

READ IN THE CELL ID  
READ IN THE CURRENT  
CELLS VARIABLE TABLE  
ADDR.  
READ IN THE NUMBER OF  
VARIABLES

RXDRAM RG1

LAR ARO, RG1

LAR ARI, CELVAR

SET UP THE POINTER TO  
THE VARIABLES

SXRAM TABLE

LARP 0

BANZ RESTLP

ARP = 0

IF THERE ARE ANY MORE  
VARIABLES SAVE THEM

B RESRET

RESTLP

LARP 1

RXDRAM \*+AR0

WRITE OUT THE NEXT  
VARIABLE

BANZ RESTLP

ARE THERE ANY MORE?

RESRET

LACKB CSASB

FIND THE RETURN  
ADDRESS FROM THE CSASB  
THE CELLS ID  
READ IN THE RETURN  
ADDRESS

ADD RG2  
TBLR RG1

LAC RG1

PUSH

RET

PUSH IT ONTO THE STACK  
AND GO THERE

05-03-02

52

\*\*\*\*\*  
\* ABORT THE CURRENTLY ACTIVE SEQUENCE AND RETURN TO  
\* CELL 0 THIS ROUTINE SHOULD BE BRANCHED TO:  
\*\*\*\*\*

ABORT

CALL SAVE

SAVE THE ENVIROMENT OF THE  
CURRENTLY ACTIVE CELL

ABORT1

LACB

CCCCB

SET THE CCB POINTER TO CELL 0  
CCB

SACL CCBSP

B

RESTOR

AND RESTORE CELL 0

\*\*\*\*\*  
\* END THIS CELL AND INSTALL THE NEXT ONE IN THE  
\* SEQUENCE THIS ROUTINE SHOULD BE BRANCHED TO:  
\*\*\*\*\*

ENDCEL

CALL SAVE

SAVE THE ENVIROMENT OF THE  
LAST CELL

\* FIND THE ADDRESS OF THE NEXT CELL IN THE SEQ

RDRAM CCBSP, CURSSQ

LAC CURSSQ

INCREMENT THE SEQ POINTER

ADD ONE

SACL CURSSQ

B

RESTOR

INSTALL THE NEXT CELL

00:03:00

\*\*\*\*\*  
THIS ROUTINE IS ONLY EVER CALLED BY CELL 0. IT  
STARTS OFF A NEW SEQUENCE.  
\*\*\*\*\*

RUN RDXDRAM CURSSQ.RGIRGI CONTAINS THE SEQUENCE  
NUMBER WHICH IS A  
DIRECT OFFSET TO THE  
SEQUENCE POINTER.  
GET THE RETURN ADDRESS  
POP  
SACL RG2  
LACKB CSASB AND WRITE IT OUT TO THE FIRST  
LOCATION IN THE  
TBLWRG2 CSASB  
B ENDCEL AND PERFORM ENDCEL WHICH  
WILL SAVE THE ENVIROMENT OF  
CELL 0 AND THEN INSTALL THE FIRST CELL IN THE SEQUENCE

\*\*\*\*\*  
PEND  
\*\*\*\*\*

03-03-02

692 32 869 6 08  
Associative Measurement PTY. LTD

06. März 2003  
S 17755 EP/DE AI/Sn/bb

### Ansprüche

1. Wissenschaftlicher Instrumentenemulator, aufweisend:  
einen Computer (1) mit einem Speicher (3), der mit einem Prozessor (2) gekoppelt ist;  
ein multifunktionales Eingangs-/Ausgangs-Signalmodul (6), welches zu-  
mindest einen analogen Signaleingang (7) mit Digitalisierungsmitteln (37)  
hierzu verbunden und zumindest einen Signalausgang (8) aufweist und  
konfigurierbar ist mit Bezug auf zumindest eine Abtastfrequenz des analo-  
gen Signaleingangs (7);  
eine Videoanzeige (10), die mit einem Videoanzeigengenerator (50) gekop-  
pelt ist, wobei der Videoanzeigengenerator (50) mit dem Prozessor (2) ge-  
koppelt ist;  
ein Bibliotheksprogramm, das in dem Speicher (3) gespeichert ist, wobei  
das Bibliotheksprogramm eine Mehrzahl von Instrumentationseinheiten  
aufweist, wobei jede geeignet ist, auf der Videoanzeige (10) angezeigt zu  
werden, und jede eine vorbestimmte Signalverarbeitungsfunktion aufweist;  
ein Set-Up-Programm, das in dem Speicher (3) gespeichert ist, und durch  
den Prozessor (2) ausgeführt wird, um einem Bediener zu erlauben, aus-  
gewählte Instrumentationseinheiten miteinander zu verbinden, um ein Feld  
von miteinander verbundenen Instrumentationseinheiten zu bilden, wobei  
eine Abtastfrequenz für das multifunktionale Eingangs-/Ausgangs-  
Signalmodul (6) spezifiziert ist, wobei das Set-Up-Programm Signalkück-  
führungsverbindungen in dem Feld ermöglicht, wobei das Feld einen Aus-  
gang aufweist, der zumindest einem der folgenden zugeführt wird: der Vi-  
deoanzeige, dem multifunktionalen Eingangs-/Ausgangssignalmodul (6)  
und dem Speicher (3); und

ein graphisches Compiler-Programm, das in dem Speicher (3) gespeichert ist und durch den Prozessor (2) ausführbar ist bei Vollendung des Set-Up-Programms, um ein wissenschaftliches Instrumentationsemulationsprogramm zu kreieren, um eine Gesamt-Signalverarbeitungsfunktion des Feldes von ausgewählten verbundenen Instrumentationseinheiten auszuführen, wobei das graphische Compiler-Programm in der Lage ist, Signalarückkopplung in dem Feld aufzulösen und eine Sequenz zu bilden in dem Emulationsprogramm der vorbestimmten Signalverarbeitungsfunktionen der Instrumentationseinheiten, die jede auszuführen ist einmal zwischen sukzessiven Samples eines Eingangssignals, das erhalten wird unter Verwendung des multifunktionalen Eingangs-/Ausgangssignalmoduls, wobei das Emulationsprogramm Operationen aufweist, die gemäß der spezifizierten Abtastfrequenz konfiguriert sind, zum Abtasten des Eingangssignals, um einen Datenstrom von dem multifunktionalen Eingangs-/Ausgangssignalmodul (6) zu schaffen;  
eine Instrumentenemulationsunteranordnung, die gekoppelt ist mit dem Prozessor (2), dem Speicher (3) und dem Videoanzeigengenerator, wobei die Instrumentenemulationsunteranordnung direkt verbunden ist mit dem multifunktionalen Eingangs-/Ausgangs-Signalmodul, wobei die Instrumentenemulationsunteranordnung aufweist:  
einen zweiten Prozessor zum Ausführen des wissenschaftlichen Instrumentenemulationsprogramms, um kontinuierlich die Gesamt-Signalverarbeitungsfunktion auszuführen, derart, dass jedes Sample des Datenstroms von sukzessiven Samples verarbeitet wird während einer einzelnen Sample-Zeit, um eine Felddausgabe des wissenschaftlichen Instrumentenemulators zu produzieren, bevor ein sukzessives Sample des Eingangssignals erhalten wird durch das multifunktionale Eingangs-/Ausgangssignalmodul (6);  
wobei zumindest eine einer Mehrzahl von Operationen durchgeführt wird, wobei die Mehrzahl von Operationen jede von Anzeigen der Felddausgabe auf der Videoanzeige (10) in Echtzeit Speichern der Felddausgabe in dem



Speicher (3) und Bereitstellen der Felddausgabe zu dem Signalausgang des multifunktionalen Eingangs-/Ausgangssignalmoduls (6) in Echtzeit umfasst.

2. Wissenschaftlicher Instrumentenemulator gemäß Anspruch 1, bei welchem die Instrumentationseinheiten, jede repräsentiert sind durch ein entsprechendes Icon, welches geeignet ist auf der Videoanzeige angezeigt zu werden während des Betriebs des Set-Up-Programms, wobei jede gewünschte Instrumentationseinheit ausgewählt werden kann, die angeordnet ist in und verbunden ist mit anderen Instrumentationseinheiten in dem Feld.

3. Wissenschaftlicher Instrumentenemulator gemäß Anspruch 2, bei welchem das Feld eine Datensignal-Rückkopplungsschleife aufweist, die zwischen einem Ausgang einer der Instrumentationseinheiten und einem Eingang einer der Instrumentationseinheiten verbunden ist.

4. Wissenschaftlicher Instrumentenemulator gemäß Anspruch 1, bei welchem das multifunktionale Eingangs-/Ausgangssignalmodul einen zusätzlichen Signalausgang aufweist zur Verbindung mit externer Hardware und welcher verfügbar ist als ein Echtzeit-elektrisches Signal, welches geeignet ist, die externe Hardware zu bedienen oder zu triggern.

5. Wissenschaftlicher Instrumentenemulator gemäß Anspruch 4, bei welchem das multifunktionale Eingangs-/Ausgangssignalmodul sowohl analoge als auch digitale Signalausgänge aufweist.

6. Wissenschaftlicher Instrumentenemulator gemäß Anspruch 4, bei welchem die Felddausgabe, die in dem Speicher gespeichert wird, verfügbar ist für nachfolgende graphische Manipulation und/oder Tabulation durch den Computer.

05.03.02

4

7. Wissenschaftlicher Instrumentenemulator gemäß Anspruch 1, bei welchem die Instrumentenemulations-Prozessoranordnung des Weiteren einen Videogenerator aufweist, der mit dem zweiten Prozessor verbunden ist, zum Bereitstellen einer anzeigbaren Repräsentation der Felddausgabe, bevor ein nachfolgendes Sample erhalten wird.
8. Verfahren zum Bestimmen der Ausführungsreihenfolge vorbestimmter Signalverarbeitungselemente in einem Computer, der eine Zentralverarbeitungseinheit (2) und einen elektronischen Speicher (3) aufweist, um in Echtzeit eine mathematische/Signalverarbeitungsfunktion eines emulierten Instruments zu implementieren, welches zumindest eine wiederholt abgelesene Signaleingabe aufweist, um davon eine Signalausgabe zu bilden, wobei das Verfahren die folgenden Schritte aufweist:
10. Anwenden eines Set-Up-Programms, welches in dem Speicher (3) gespeichert wird und durch die Zentralverarbeitungseinheit (2) ausgeführt wird, um einem Bediener zu erlauben, ausgewählte Instrumentationseinheiten miteinander zu verbinden, wobei eine Abtastfrequenz für ein multifunktionales Eingangs-/Ausgangssignalmodul (6) spezifiziert wird, wobei das Set-Up-Programm Rückkopplungsverbindung in dem Feld ermöglicht, wobei
15. das Feld einen Ausgang aufweist, welcher zumindest einem der folgenden zugeführt wird: einer Videoanzeige (10), dem multifunktionalen Eingangs-/Ausgangssignalmodul (6) und dem Speicher (3), wobei jede der Instrumentationseinheiten eine mathematische/Signalverarbeitungsfunktion aufweist;
20. Kreieren eines wissenschaftlichen Instrumentenemulationsprogramms unter Verwendung eines Compiler-Programms, welches in dem Speicher (3) gespeichert wird und durch die Zentralverarbeitungseinheit (2) bei Vollendung des Set-Up-Programms ausführbar ist, wobei das wissenschaftliche Instrumentenemulationsprogramm die Gesamt-Signalverarbeitungsfunktion des Feldes ausgewählter verbundener Instrumentationseinheiten ausführt.
25. 30.

Bestimmen einer Ausführungsreihenfolge der mathematischen/Signalverarbeitungsfunktion mit den folgenden Schritten:

Repräsentieren der Funktion als eine Sequenz elementarer mathematischer Schritte, die ihrerseits repräsentierbar sind unter Verwendung vordefinierter Signalverarbeitungselemente; und

Anordnen der vorbestimmten Signalverarbeitungselemente in der Sequenz für sequentielle Ausführung innerhalb der Zeit zwischen sukzessiven Samples und Beginnen mit der zumindest einen abgetasteten Signaleingabe,

wobei der Bestimmungsschritt hierdurch ein Feld von Instrumentationseinheiten bildet, welches die Gesamtschaltungsmathematische/Signalverarbeitungsfunktion aufweist;

Ausführen der Gesamt-Signalverarbeitungsfunktion, derart, dass jedes Sample des zusammenhängenden Datenstroms verarbeitet wird während einer einzelnen Abtastzeit, um eine Feldausgabe der gesamtmathematischen Signalverarbeitungsfunktion zu produzieren, bevor ein sukzessives Sample des Eingangssignals erhalten wird durch das multifunktionale Eingangs-/Ausgangssignalmodul (6);

Repräsentieren der Gesamtschaltungsmathematischen/Signalverarbeitungsfunktion als eine Sequenz von Ereignissen, die darstellbar sind als vorbestimmte Signalverarbeitungselementereignisse;

Anordnen der vorbestimmten Signalverarbeitungselementereignisse in der Sequenz für sequentielle Ausführung, beginnend mit der zumindest einen abgetasteten Feldsignaleingabe, wobei eine bestimmte Rechnerzeit der vorbestimmten Signalverarbeitungselemente gesichert wird durch Ausführen jedes der vorbestimmten Signalverarbeitungselemente einmal zwischen sukzessiven Samples, wobei das Verfahren in der Lage ist, Datensignal-Rückkopplung in dem Feld von Instrumentationseinheiten aufzulösen;

Anzeigen der Feldausgabe auf der Videoanzeige (10) in Echtzeit;

Speichern der Feldausgabe in dem Speicher (3); und  
Bereitstellen der Feldausgabe an den Signalausgang des multifunktionalen  
Eingangs-/Ausgangssignalmoduls in Echtzeit.

5 9. Verfahren gemäß Anspruch 8, bei welchem ein Parameter des funktionalen  
Blocks spezifiziert werden kann.

10. Verfahren gemäß Anspruch 8, bei welchem das Feld zumindest eine Da-  
tensignal-Rückkopplungsschleife umfasst, in welcher ein Ausgang von ei-  
10 ner der Instrumentationseinheiten verbunden ist, um einen Eingang von ei-  
ner der Mehrzahl von Instrumentationseinheiten zu bilden, und ein ur-  
sprüngliches Ergebnis eines entsprechenden der ausführbaren Ereignisse in  
der Sequenz ausführbarer Ereignisse verwendet wird in einer wiederholten  
Ausführung eines früheren Ereignisses, um ein modifiziertes Ergebnis des  
15 entsprechenden einen der ausführbaren Ereignisse von einer vorherigen  
Abtastperiode zu erzeugen.

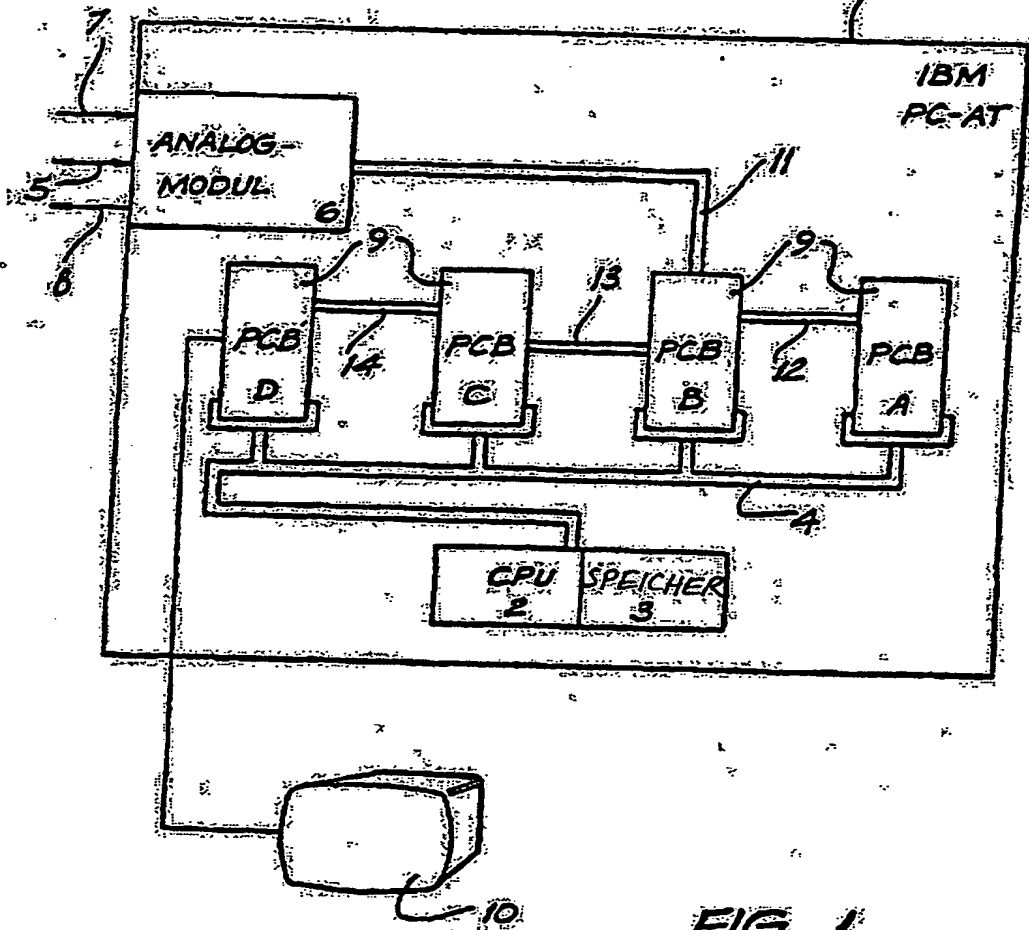


FIG. 1

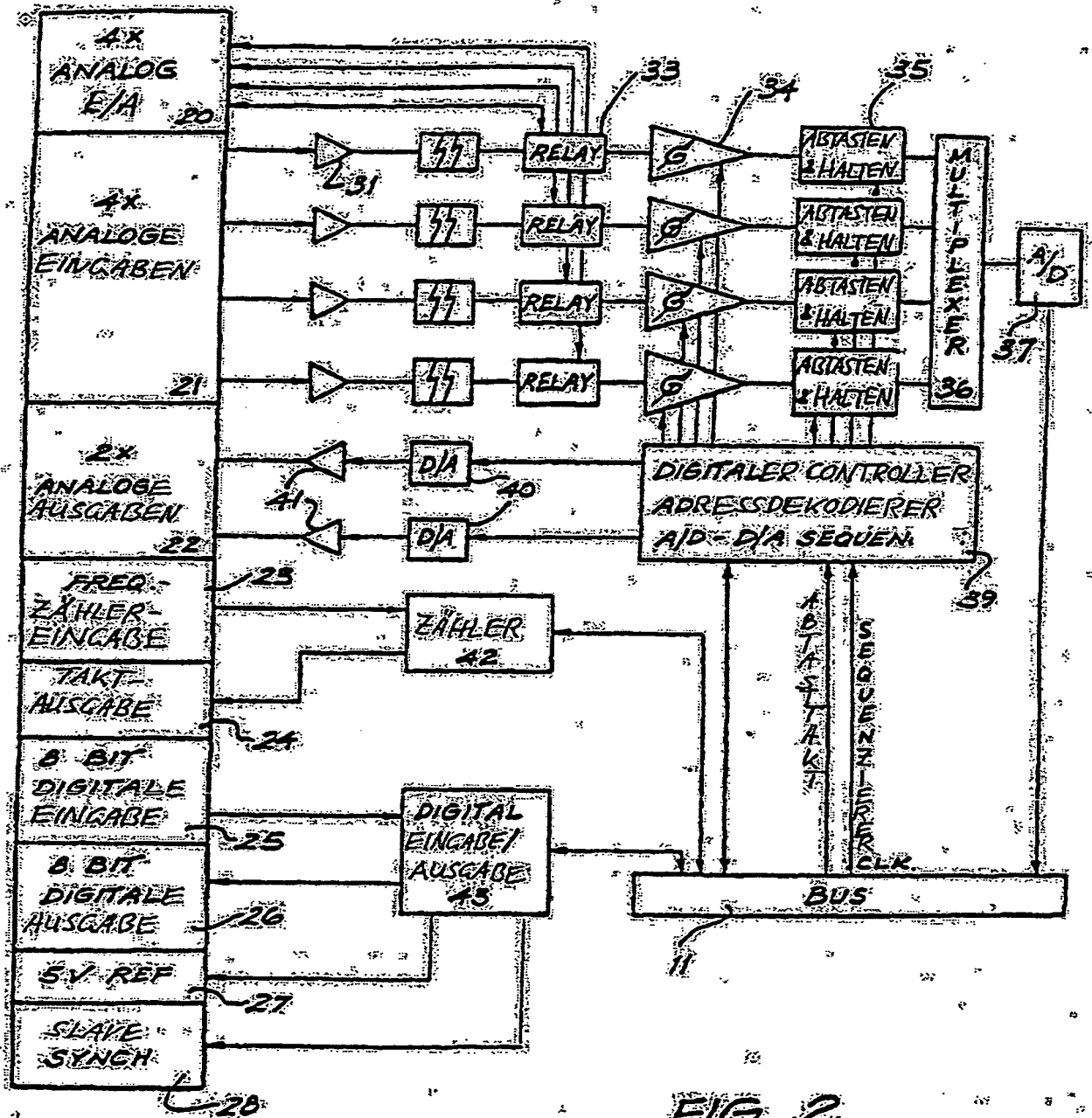


FIG. 2

00:03:00

3/15

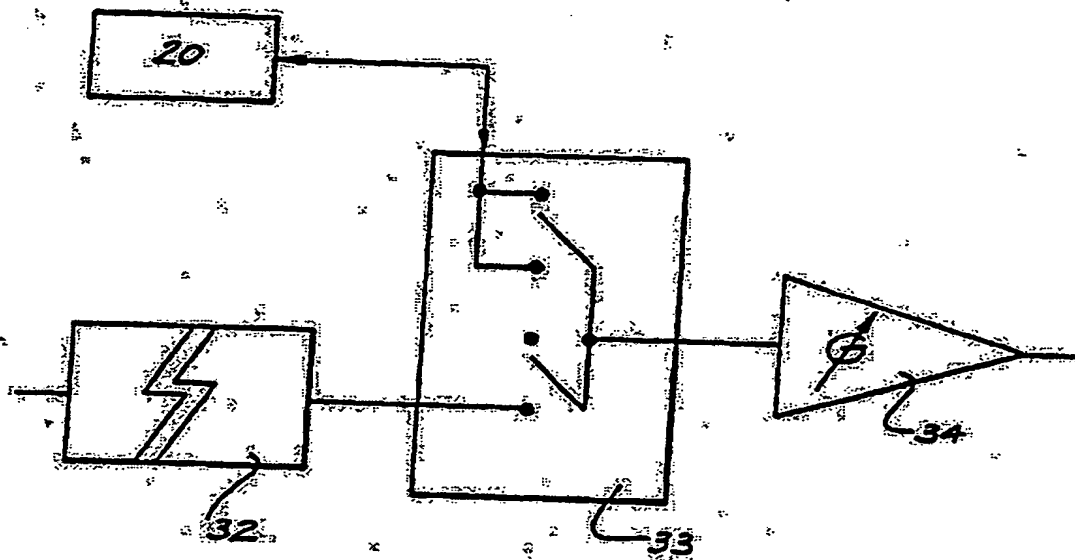
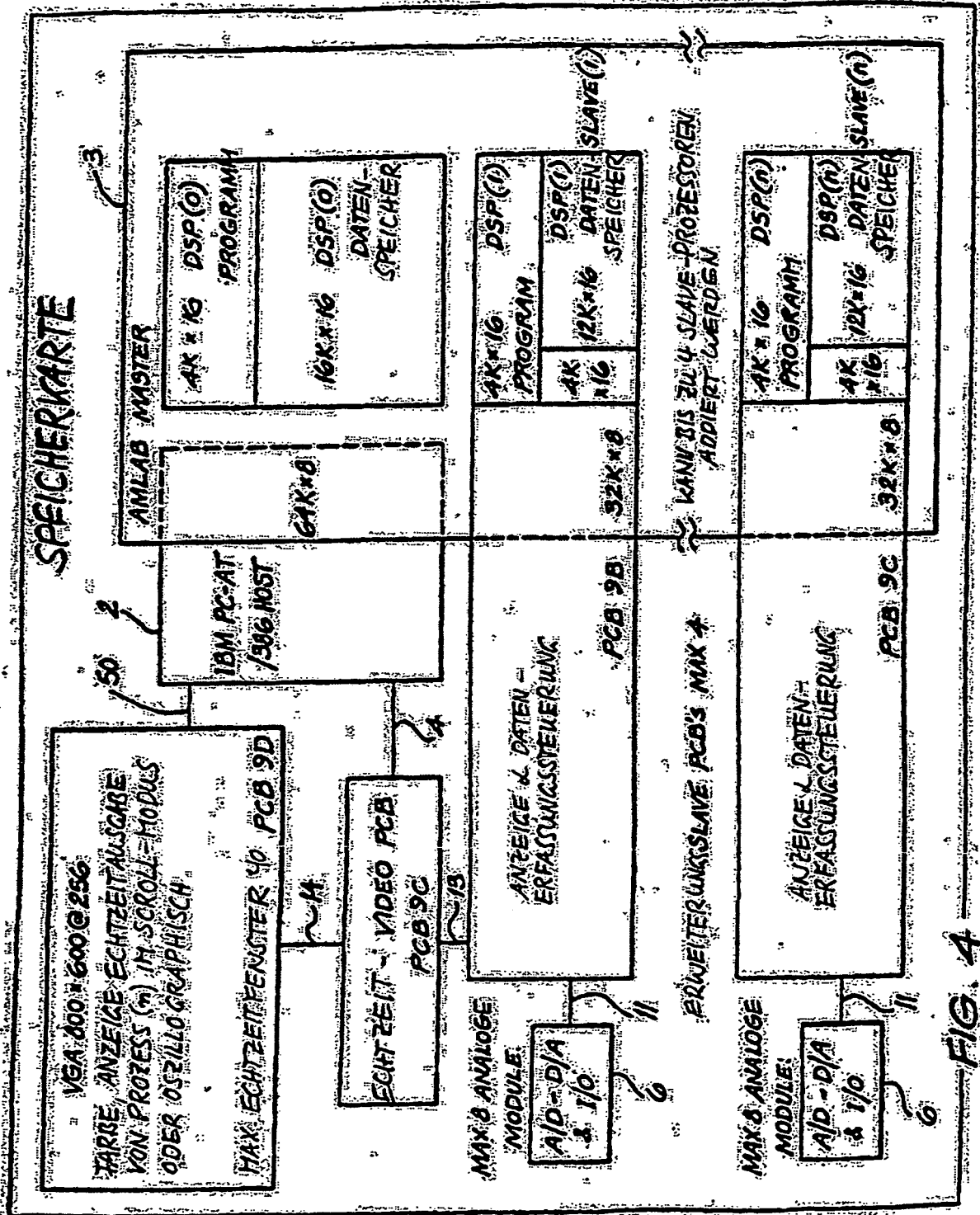
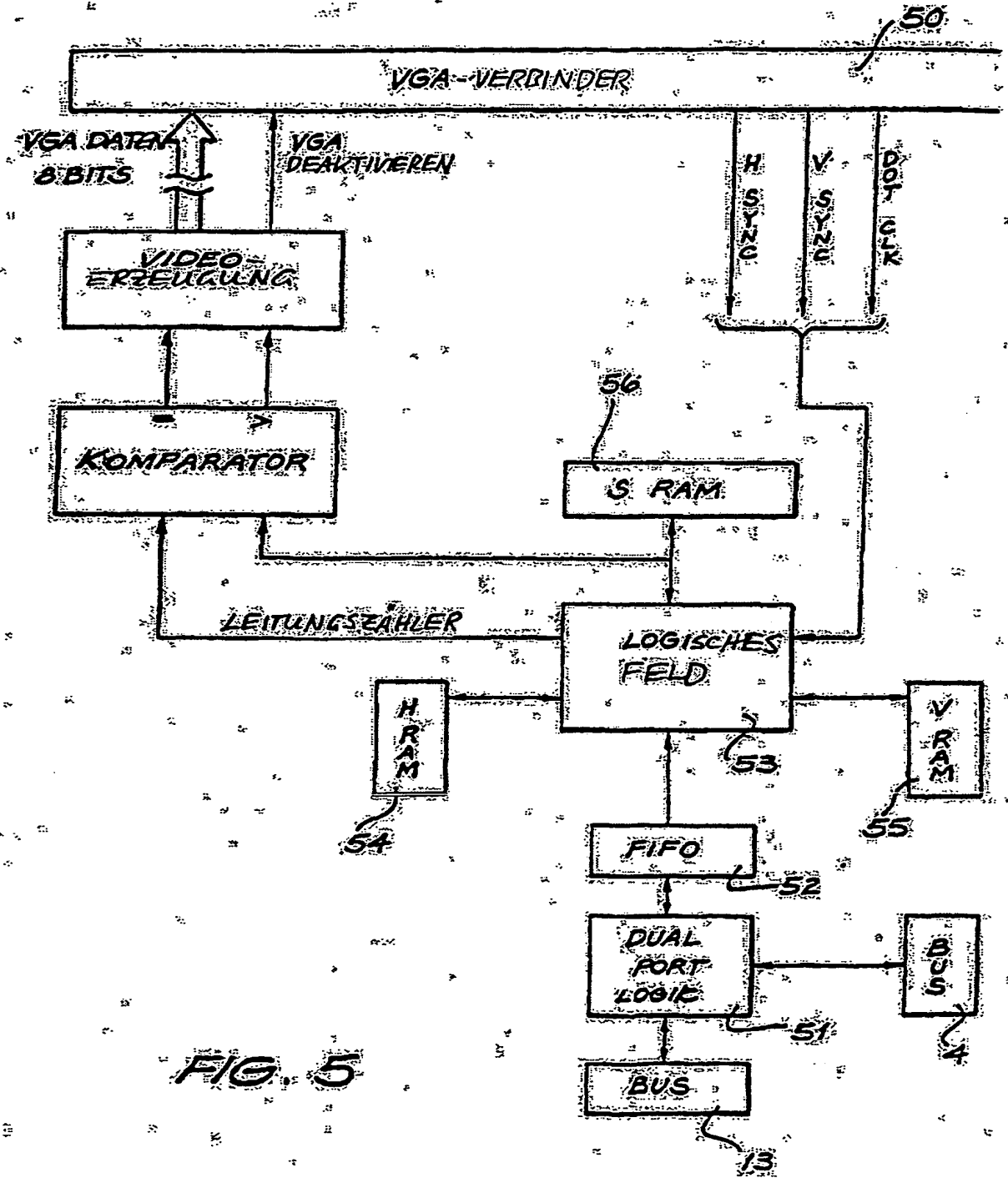


FIG. 3







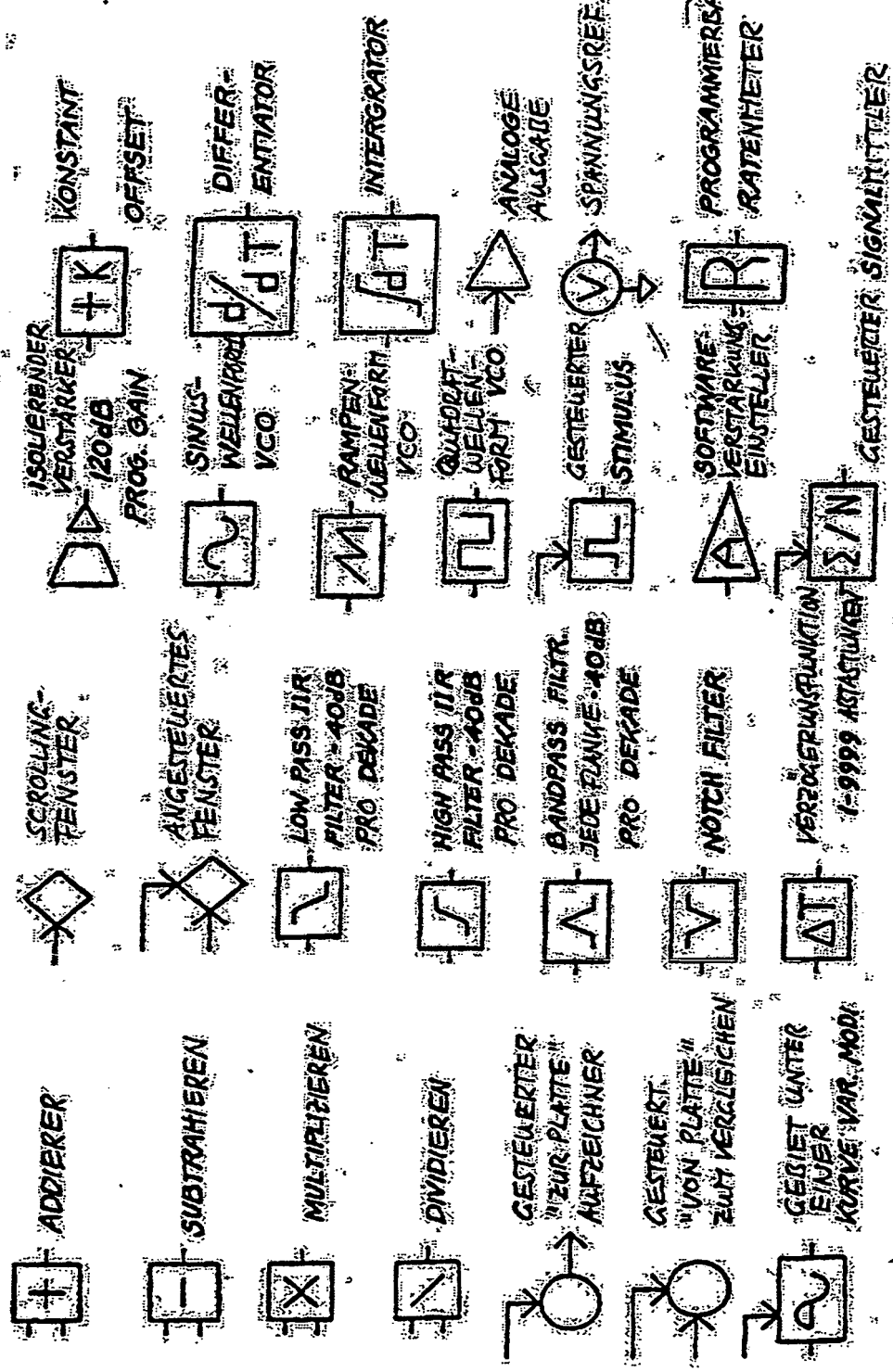


FIG. 6

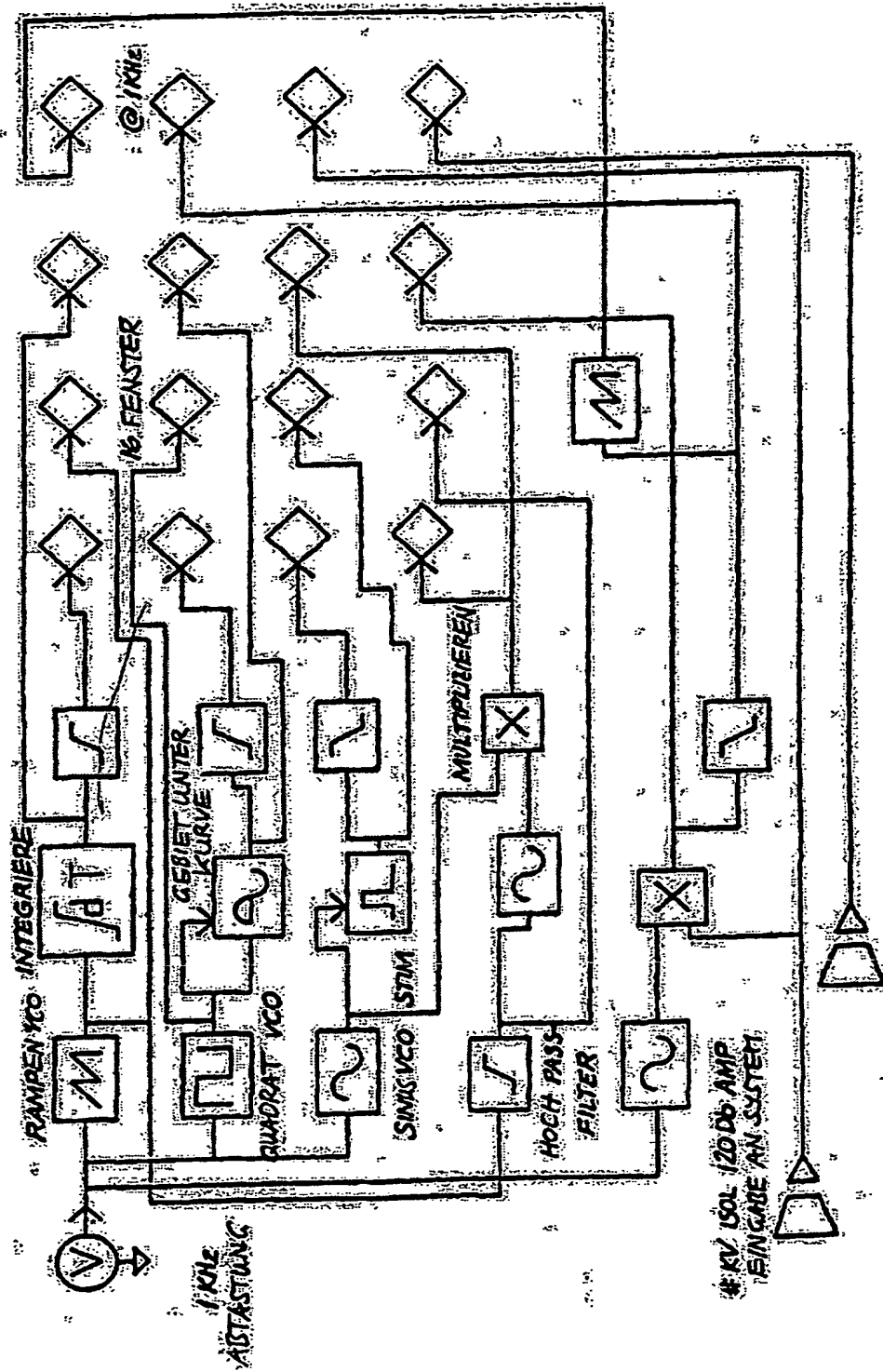
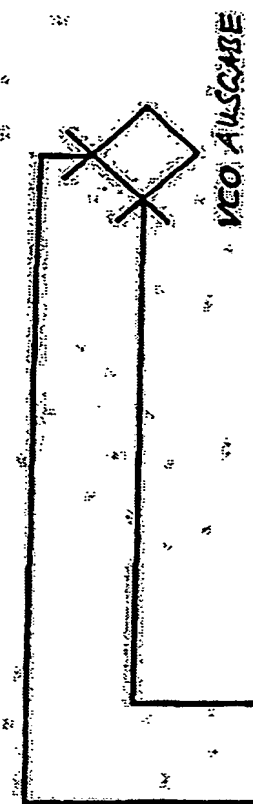


FIG. 7



0 VOLT

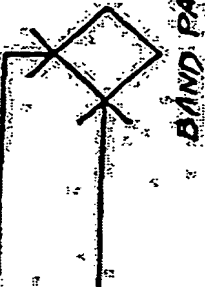
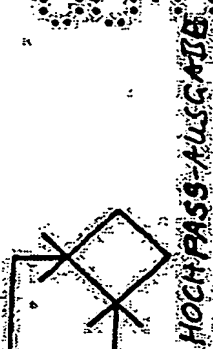
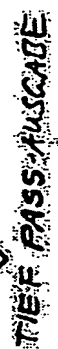


FIG. 8

8/15

9/15

08.03.02

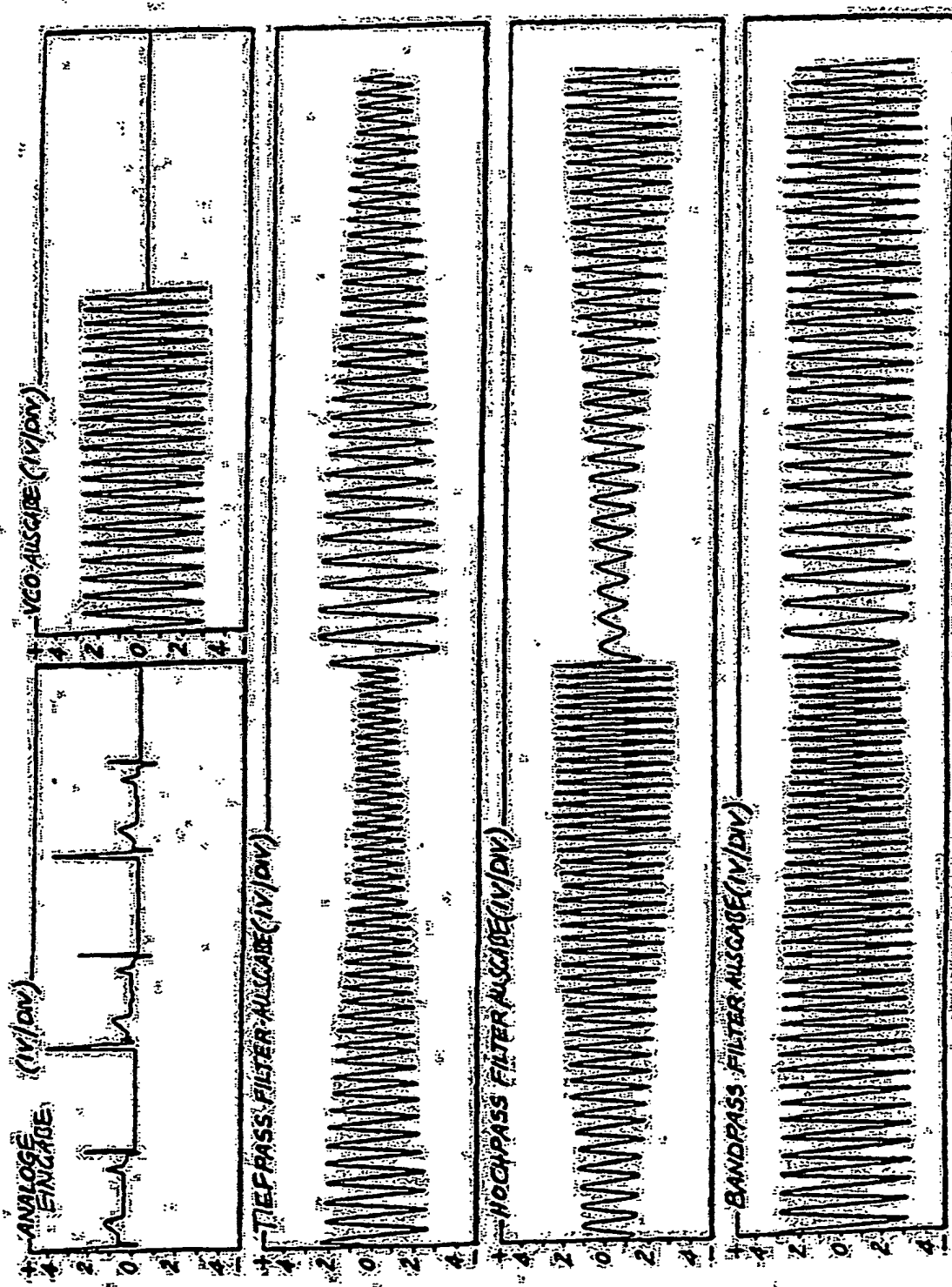


FIG. 9



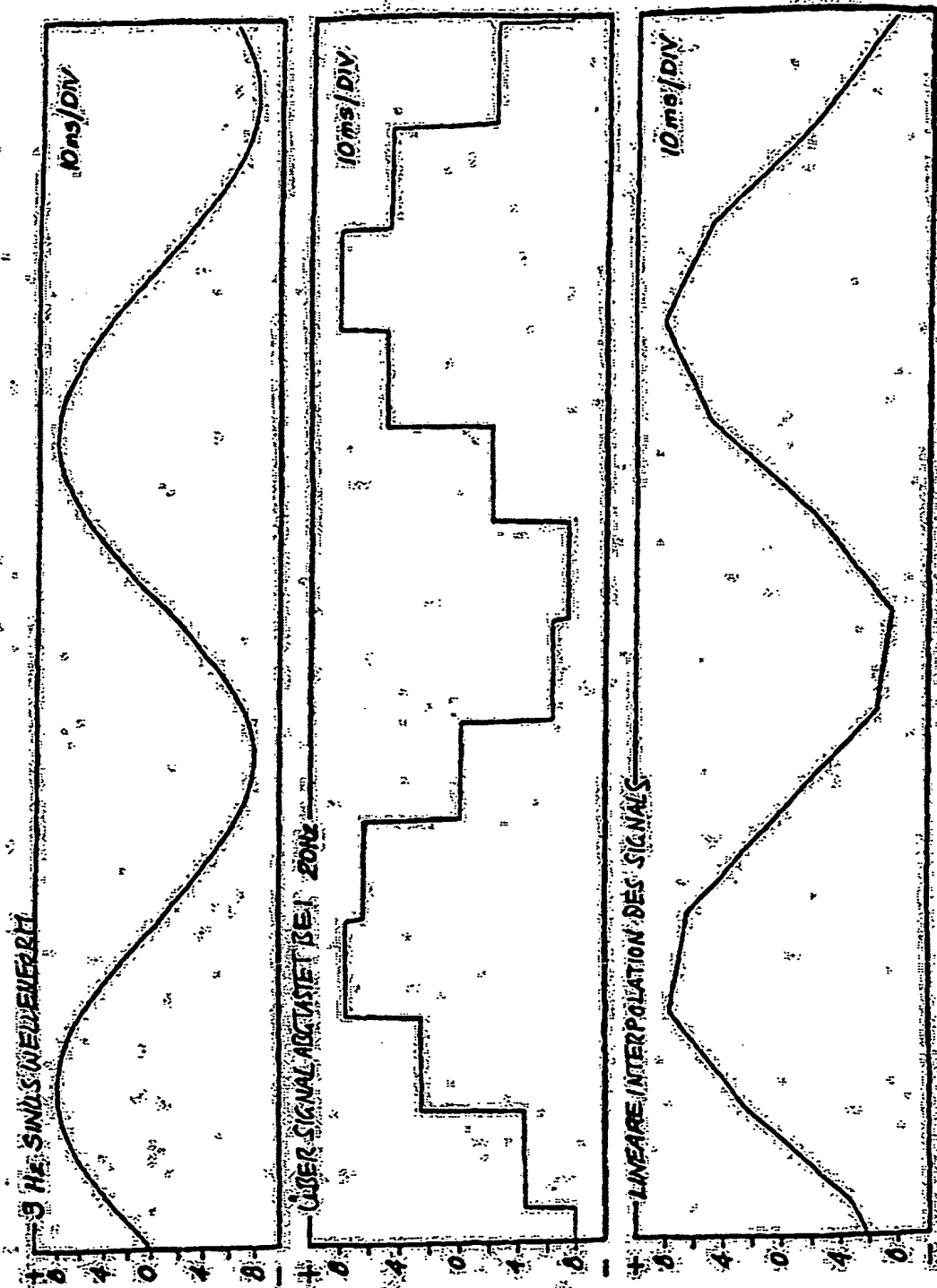


FIG. 11

08.03.03

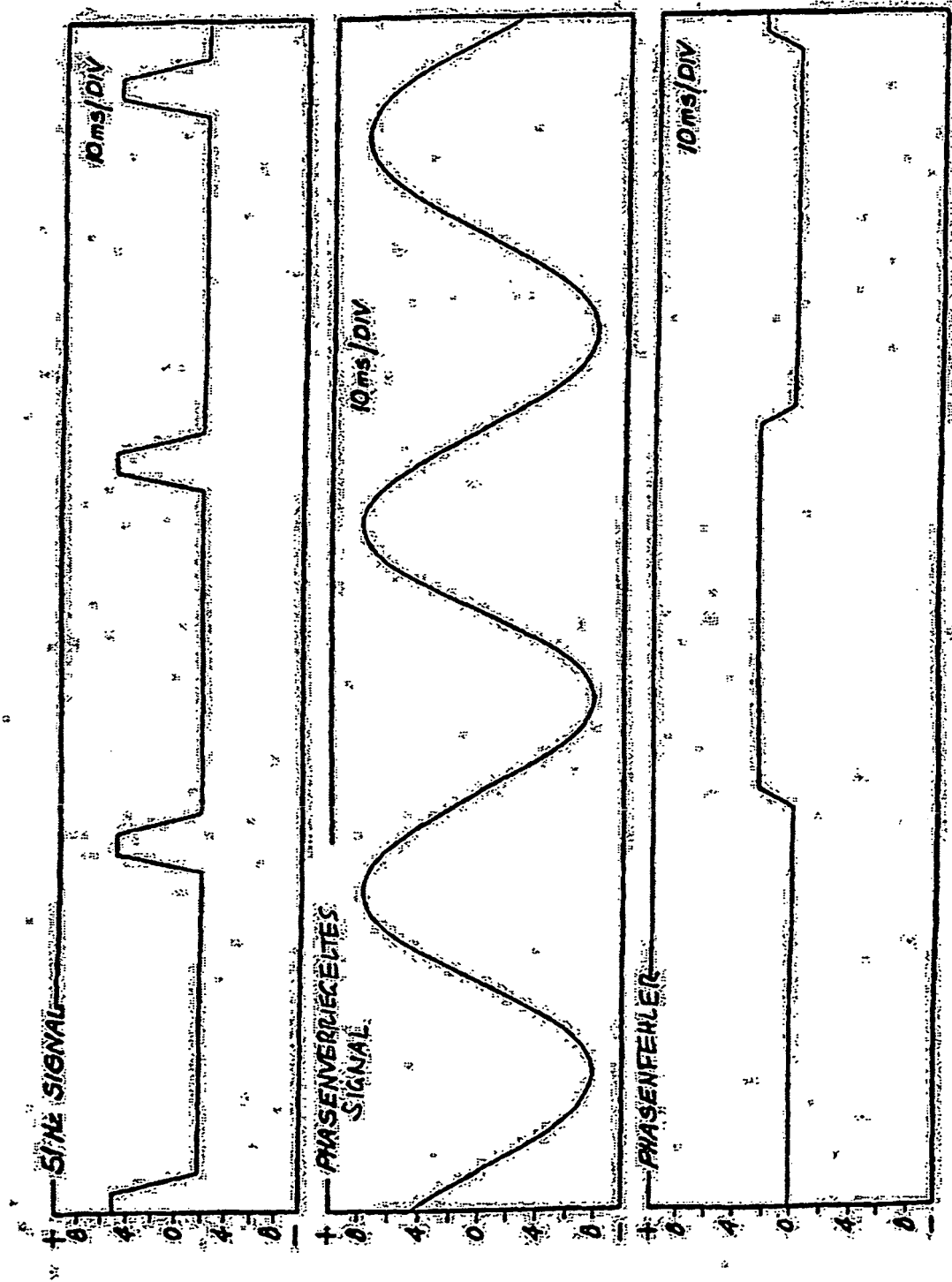


FIG. 13



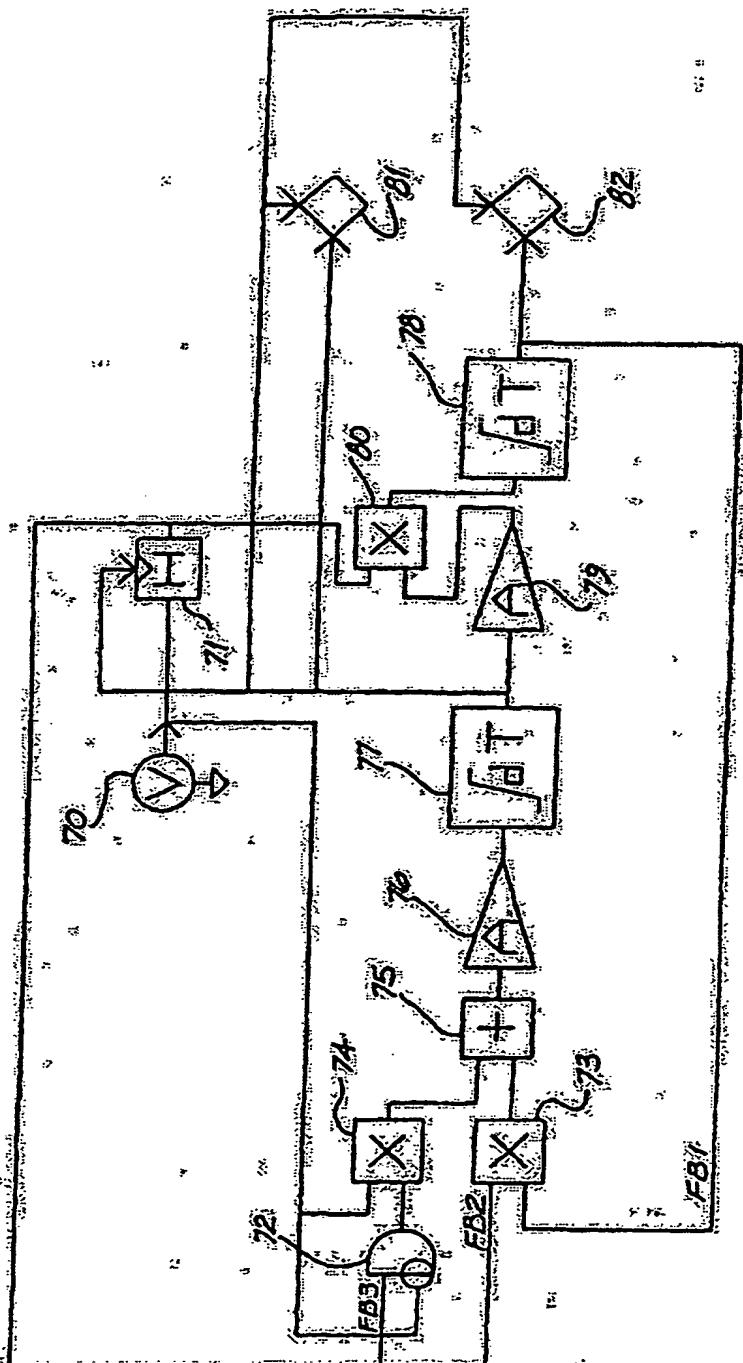


FIG. 1A

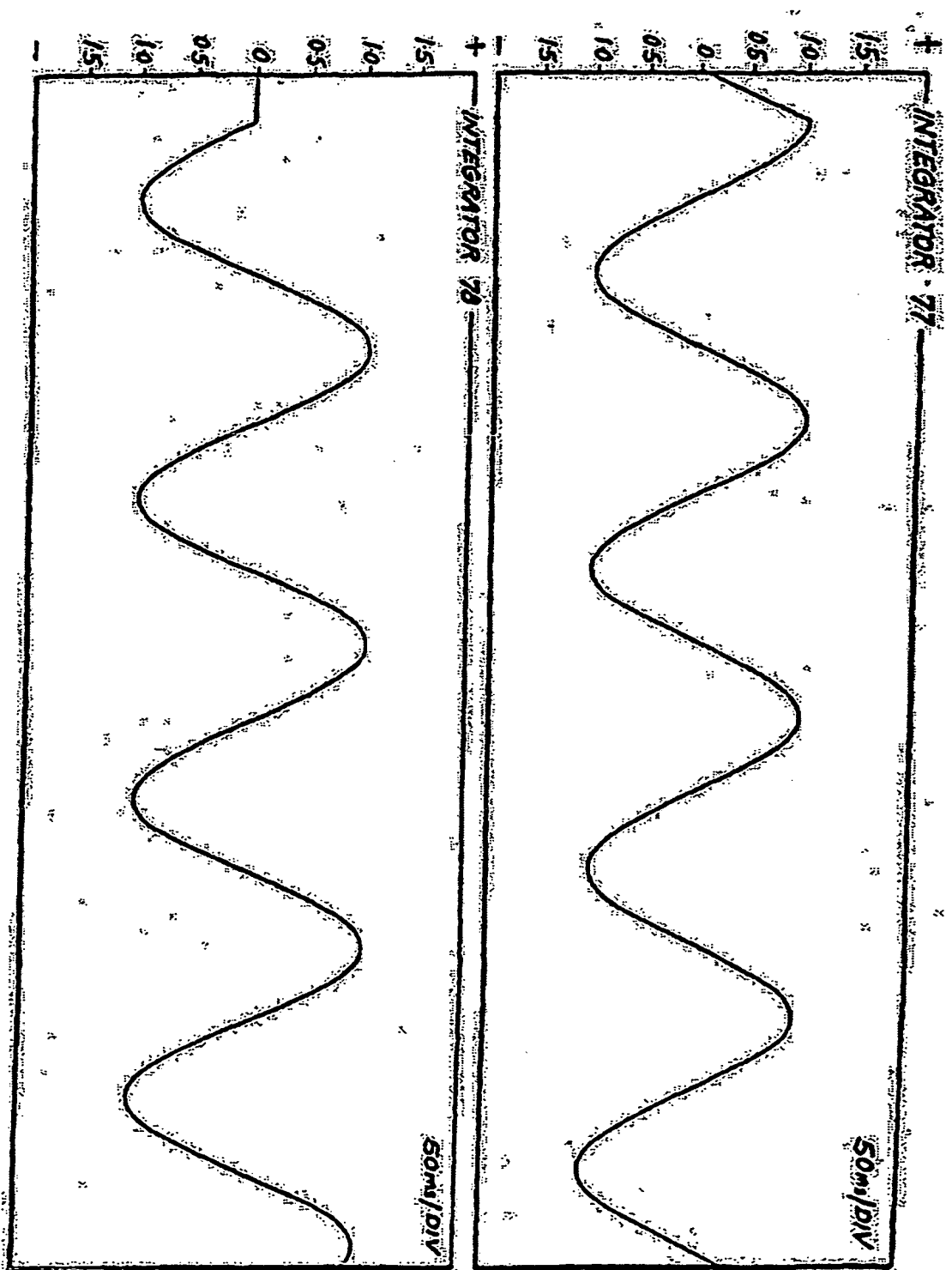


FIG. 15

00000000

00-00-00

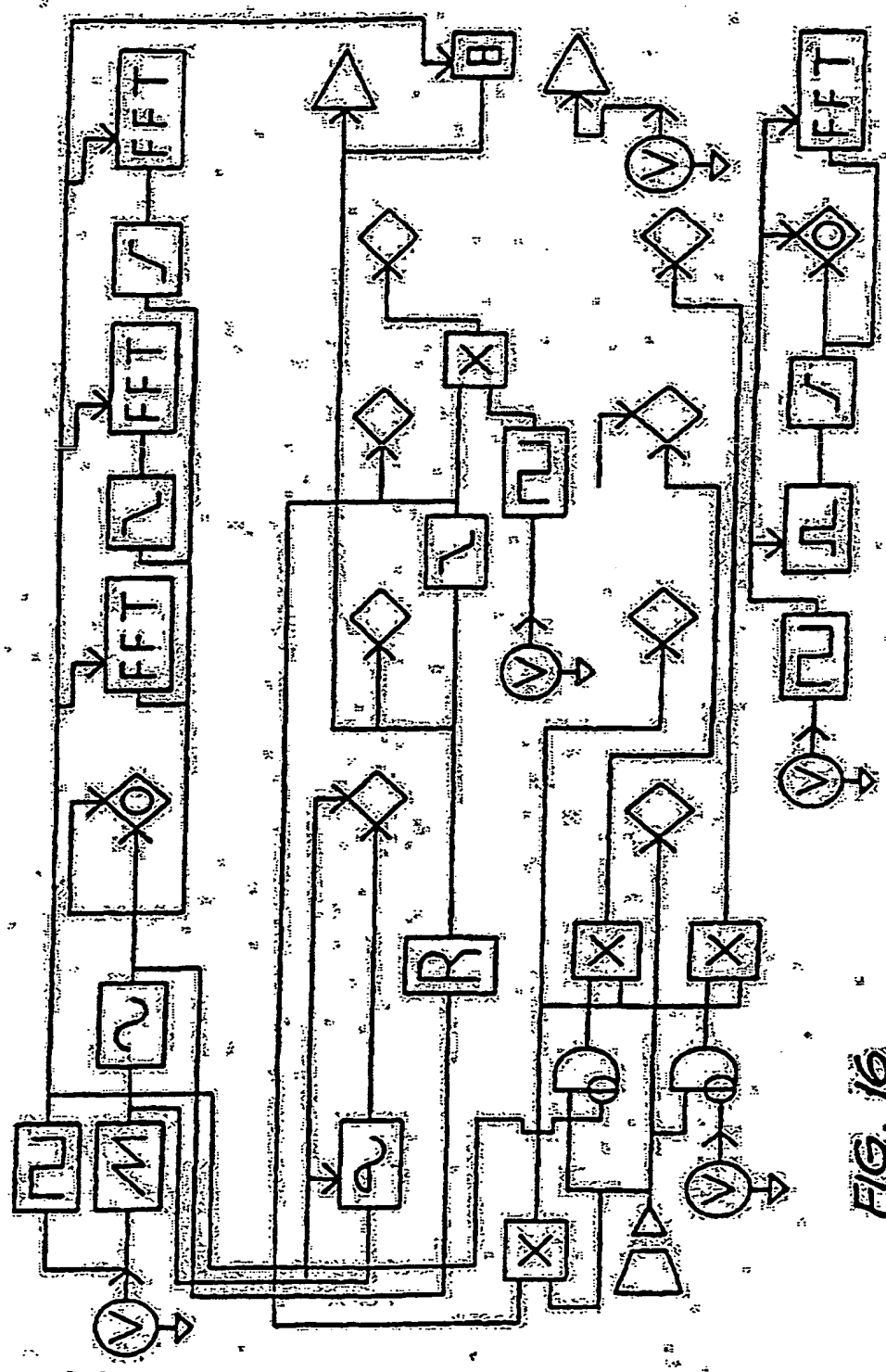


FIG. 16



**THIS PAGE BLANK (USPTO)**

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☒ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

**THIS PAGE BLANK (USPTO)**